WHITEPAPER

Building Compliance In, Not On

ISO 27001, SOC 2, GDPR and the Vibe-Coder Gap

Why security, privacy and compliance standards must be designed into a product from day one — and why the fastest builders are often the ones most exposed to the risk of ignoring them.

By Justin James
justinjames.agencie.io

Prepared for engineering leaders, founders, developers and compliance stakeholders
Published Monday, 13 July 2026, 09:48 (ICT, UTC+7)

Contents

1. Executive Summary

The way software gets built has changed faster than the way it gets governed. A single developer with an AI coding assistant can now stand up a working product — authentication, a database, payments, a public API — in an afternoon. That velocity is a genuine advantage. But the standards that determine whether a product is *trustworthy* — ISO/IEC 27001, SOC 2, the GDPR, and their peers — were written for a world in which security and privacy were deliberate, reviewed, documented decisions. The gap between how fast we can now build and how carefully these standards expect us to build is the central risk this paper addresses.

We call the person operating at the sharp end of that gap the **"vibe coder"**: someone who builds by intuition and momentum, often with heavy AI assistance, shipping what *feels* right without a working model of the invisible constraints — data-handling obligations, access controls, audit trails, retention rules, breach-notification duties — that a regulated or enterprise-grade product must satisfy. The vibe coder is not careless or unskilled. They are simply optimising for the thing they can see (a feature that works) over the things they cannot (the controls an auditor, a regulator, or an attacker will eventually test).

This paper makes three arguments. First, that these standards are not bureaucratic overhead but encoded lessons from real failures, and treating them as design inputs is cheaper and safer than treating them as an afterthought. Second, that **retrofitting** compliance onto a product built without it is disproportionately expensive, sometimes requiring re-architecture. And third, that the vibe-coder gap is closeable — not by slowing builders down, but by making the invisible constraints visible early, through the right defaults, tooling, and a shared vocabulary.

> **The one-sentence version**
>
> Speed and compliance are not opposites — but only if the constraints are built into the product's design, its defaults, and its developers' mental models before the first line of production code, rather than discovered during an audit, a due-diligence review, or a breach.

2. The Vibe-Coder Gap

Modern development tooling collapses the distance between an idea and a running system. Scaffolding, boilerplate, infrastructure and integrations that once took a specialised team can now be generated on demand. This is empowering, and it has pulled a much wider range of people into building real products. It has also detached the act of building from the accumulated context that used to travel with it — the code review, the threat-modelling session, the security architect who asked "where does this data go, and who can see it?"

The result is a specific and recurring pattern. The product works. The demo is flawless. Customers sign up. And then a prospect's security team sends a due-diligence questionnaire, or a customer in the EU asks how their personal data is processed, or an enterprise buyer asks for a SOC 2 report — and the team discovers that the product has no concept of an audit log, secrets are committed in plaintext, every engineer has production database access, personal data is scattered across systems with no retention policy, and there is no documented way to delete a user. None of these were *bugs*. The software did exactly what it was told. The problem is that nobody told it to satisfy constraints it could not see.

Why AI assistance widens the gap

AI coding assistants are trained to produce code that satisfies the prompt. Ask for a login system and you get a login system that authenticates users. You will not, unless you explicitly ask, get rate limiting, brute-force lockout, secure session management, structured audit logging, least-privilege database roles, or a data-retention strategy for the credentials being stored. The assistant is not wrong — it answered the question. But compliance lives almost entirely in the parts of the problem the prompt did not mention. The faster and more fluent the assistant, the more confidently a builder can ship something that is functionally complete and structurally non-compliant.

> **What the vibe coder cannot see**
>
> Access boundaries · audit and change logs · data classification and retention · encryption in transit and at rest · secrets management · least-privilege roles · lawful basis for processing personal data · the right to erasure · breach-notification timelines · vendor/sub-processor risk · evidence that any of the above is actually happening.

3. What “Building It In” Actually Means

Every major standard converges on the same underlying idea, expressed in different vocabularies: **security and privacy are properties of a system's design, not features added to it later**. The GDPR calls this “data protection by design and by default.” ISO 27001 calls it an Information Security Management System (ISMS) — a set of processes woven through the organisation. SOC 2 calls it a control environment. NIST frames it as security engineering. The words differ; the demand is identical: decisions about who can access what, how data is protected, how long it is kept, and how you would prove any of it must be made *as the system is designed*, because they are extremely hard to bolt on afterwards.

Concretely, building it in means the architecture itself encodes the constraints. Data is classified so the system knows what is sensitive. Access is least-privilege by default, so a new service or employee starts with nothing and is granted only what they need. Every consequential action writes an immutable audit record, because you cannot invent that history retroactively. Personal data has a defined lifecycle — a lawful reason to collect it, a place it lives, and an automated point at which it is deleted. Secrets live in a managed store, never in source. These are design decisions, and their cost is low when made early and punishing when made late.

4. Before You Start: The Questions to Ask First

Almost every expensive compliance problem in this paper traces back to a question that was never asked at the outset. The most valuable moment to think about security and privacy is not before an audit or a funding round — it is **before the first commit**, when the answers are still cheap to act on. The questions below take an afternoon to work through and will shape your data model, your architecture, and which obligations you inherit. Ask them at project kickoff, write down the answers, and revisit them whenever the product's scope changes.

Ask at kickoff	What the answer decides
What personal or sensitive data will this touch — and do we truly need it?	Your data-minimisation posture and GDPR exposure. The cheapest data to protect is the data you never collect.
Who are our customers, and what will they	Which frameworks become revenue gates — enterprise

Ask at kickoff	What the answer decides
demand?	buyers push SOC 2 / ISO 27001; health means HIPAA; card data means PCI DSS.
Which countries and markets will we operate in?	Which laws apply. EU/EEA users trigger the GDPR; California triggers CCPA/CPRA; each market may add its own rules.
Will the product make automated decisions about people, or use AI in a regulated way?	Whether the EU AI Act's high-risk obligations apply — data governance, human oversight and transparency you must design in.
Who will need access to what, and how do we enforce least privilege from day one?	Your access-control model. Retrofitting least privilege onto a system where everyone has everything is painful.
How will we know who did what?	Whether you have an audit trail. It cannot be reconstructed later, and both SOC 2 Type II and breach response depend on it.
What is the lifecycle of each data type — how long do we keep it, and how do we delete it?	Your retention and erasure design. “Delete this user everywhere” must be architected, not bolted on.
Who owns security and compliance — even part-time — and what's the budget?	Whether any of this actually happens. Unowned compliance is aspirational compliance.
Which third parties and sub-processors will touch our data?	Your vendor-risk surface and the data-processing agreements you'll need.
Which single framework unlocks our nearest revenue, and by when?	Your sequencing. Pursue the one that opens the next market first, rather than everything at once.

> **The point of asking early**
>
> None of these questions require slowing down or hiring a compliance team. They require ten minutes of thought before decisions harden into schemas and architecture. A team that can answer them at kickoff will build the right foundation almost by accident; a team that cannot is, without knowing it, choosing a future re-architecture.

5. ISO/IEC 27001: The Management-System Standard

ISO/IEC 27001 is the international standard for information security management. Its current edition, **ISO/IEC 27001:2022**, is now the only version against which organisations can be certified: the transition period for the previous 2013 edition ended on 31 October 2025, after which 2013-based certificates ceased to be valid. Any organisation pursuing or holding certification today is working to the 2022 standard.

The heart of the standard is not its control list but its **management system**. ISO 27001 requires an organisation to define the scope of what it is protecting, assess its information-security risks systematically, decide how to treat those risks, assign ownership, and continually improve — the familiar plan-do-check-act cycle. Certification is awarded by an accredited body only after an audit confirms the management system genuinely operates, not merely that a document exists.

Supporting the management system is **Annex A**, the reference set of controls. The 2022 revision restructured these into **93 controls organised under four themes**, a consolidation of the 114 controls and

14 domains in the 2013 edition. Eleven controls are entirely new, reflecting how the threat landscape has shifted — including threat intelligence, information security for cloud services, secure coding, data masking, data-leakage prevention, and web filtering.

Annex A theme	Controls	Representative concerns
Organizational	37	Policies, roles, supplier & cloud risk, incident management, classification
People	8	Screening, awareness, responsibilities, offboarding
Physical	14	Facility access, equipment security, secure disposal, monitoring
Technological	34	Access control, cryptography, logging, secure coding, backup, DLP

> **What it demands of a builder**
>
> You cannot certify a product in isolation — you certify an organisation's management of information security. But the technological controls map directly onto engineering: enforce access control, encrypt sensitive data, log security-relevant events, manage secrets, apply secure-coding practices, and back up recoverably. Build these in and the audit largely documents what already exists; skip them and the audit becomes a remediation project.

6. SOC 2: Proving Your Controls to Customers

SOC 2 is a reporting framework developed by the American Institute of Certified Public Accountants (AICPA). Where ISO 27001 certifies a management system, a SOC 2 report is an **independent attestation** — produced by a licensed CPA firm — describing a service organisation's controls and how well they meet the **Trust Services Criteria**. A precise but widely-muddled point: SOC 2 is an **attestation, not a certification** — there is no such thing as being "SOC 2 certified," only holding a current SOC 2 report. It is nonetheless the artefact enterprise buyers most commonly demand before entrusting a vendor with their data.

SOC 2 is built on five Trust Services Criteria. Only the first is mandatory; the others are included based on what the service actually promises its customers.

Trust Services Criterion	What it covers
Security (Common Criteria)	Mandatory. Protection of systems and data against unauthorized access — the baseline every report includes.
Availability	The system is available for operation and use as committed (uptime, resilience, disaster recovery).
Processing Integrity	System processing is complete, valid, accurate, timely and authorized.
Confidentiality	Information designated as confidential is protected across its lifecycle.
Privacy	Personal information is collected, used, retained, disclosed and disposed of per commitments.

Type I versus Type II

A **Type I** report assesses whether controls are suitably *designed* at a single point in time — a snapshot. A **Type II** report is the one buyers really want: it evaluates whether those controls *operated effectively over a period*, typically three to twelve months. This distinction is decisive for the vibe coder, because a Type II report cannot be produced retroactively. If your system was not logging access, enforcing reviews, and running its controls *throughout* the observation window, there is no evidence to audit. You cannot cram for SOC 2 the week before — the controls must have been running the whole time.

7. The GDPR: Privacy as a Legal Obligation

The EU General Data Protection Regulation, in force since 25 May 2018, is not a voluntary framework — it is law, and it reaches any organisation anywhere that offers goods or services to, or monitors the behaviour of, people in the EU/EEA (as well as anyone with an EU establishment). Unlike ISO 27001 or SOC 2, you do not choose to adopt it; if you target or track EU users, it already governs you. Its enforcement teeth are significant: fines can reach **€20 million or 4% of total worldwide annual turnover, whichever is higher**.

The GDPR is built on principles that read almost as engineering requirements. Personal data must be processed lawfully, fairly and transparently. It must be collected for specified purposes and not repurposed arbitrarily (purpose limitation). Only the data actually needed may be collected (data minimisation). It must be accurate, kept no longer than necessary (storage limitation), and protected with appropriate security (integrity and confidentiality). And crucially, the organisation must be able to demonstrate all of this (accountability).

Privacy by design and by default

Article 25 makes the paper's central thesis a legal requirement: data protection must be built into processing activities **by design and by default**. The most privacy-protective settings must be the defaults, not opt-ins. This is precisely the discipline the vibe coder is most likely to skip — the default of collecting everything, keeping it forever, and exposing it broadly is the opposite of what the law requires.

The regulation also grants individuals enforceable rights that a product must be built to honour: the right to access their data, to have it corrected, to have it erased (the “right to be forgotten”), to restrict or object to processing, and to receive their data in a portable format. Each of these implies real engineering. “Delete this user and everything derived from them, across every system and backup” is a demanding architectural requirement — trivial if designed in, and enormously painful to retrofit onto a system that assumed data would live forever in a dozen unlinked places.

> **Two obligations that catch teams off guard**
>
> **Breach notification:** a qualifying personal-data breach must generally be reported to the supervisory authority within **72 hours** of becoming aware of it — which is impossible without monitoring and logging already in place. **Records & DPIAs:** you must maintain records of processing activities and, for higher-risk processing, carry out a Data Protection Impact Assessment before you build.

8. The Wider Landscape of Design Standards

ISO 27001, SOC 2 and the GDPR are the most commonly encountered, but they sit within a broader ecosystem. Which apply depends on your industry, your data, and your customers — and it is common to be subject to several at once.

Standard	Applies when	Core demand on design
PCI DSS (v4.0.1)	You store, process or transmit payment-card data	Segment the cardholder environment, encrypt card data, restrict and log all access
HIPAA	You handle US protected health information	Administrative, physical and technical safeguards; access controls; audit trails
NIST CSF 2.0 / 800-53	US government, contractors, or as a best-practice baseline	Govern, Identify, Protect, Detect, Respond, Recover — CSF 2.0 (2024) added Govern as a sixth function
CCPA / CPRA	You handle personal data of California residents (and, increasingly, other US-state laws)	Consumer rights to access, delete and opt out; disclosure and data-handling duties
ISO 27701	You want a privacy extension to an ISO 27001 ISMS	Privacy Information Management controls layered onto 27001

The reassuring news is that these standards overlap heavily. Access control, encryption, logging, and change management appear in all of them. A team that builds these foundations well satisfies most of what every framework asks; the differences are largely scope and evidence, not fundamentally different engineering.

9. Governing the AI Itself

There is a pointed irony at the centre of this paper: the same AI that accelerates the vibe coder is itself becoming one of the most heavily regulated components a product can contain. Building *with* AI and building AI *into* a product both create obligations that did not exist a few years ago, and a builder who reaches for an AI assistant without understanding them is exposed twice over.

The EU AI Act

The EU AI Act is the world's first comprehensive AI law, and it takes a **risk-based** approach: the more consequential the use, the heavier the obligations. It sorts systems into four tiers — **unacceptable** (banned outright, such as social scoring and most real-time biometric identification), **high-risk** (permitted but tightly controlled — think AI in hiring, credit, healthcare or critical infrastructure), **limited-risk** (transparency duties, such as telling users they are talking to a chatbot), and **minimal-risk** (largely unregulated). Its penalties exceed even the GDPR's: up to **€35 million or 7% of global annual turnover**, whichever is higher.

It also applies on a phased timeline that is already partly in force, so it is not a future problem. The dates below are the ones a product team needs on its calendar.

Date	What takes effect
1 August 2024	The Act entered into force — the clock started.

Date	What takes effect
2 February 2025	Bans on unacceptable-risk practices became enforceable.
2 August 2025	Obligations for general-purpose AI (GPAI) models began — documentation, copyright, systemic-risk assessment.
2 August 2026	Core high-risk system requirements apply: risk management, data governance, logging, human oversight.
2 August 2027	Full application, including AI embedded as a safety component in regulated products.

ISO/IEC 42001 and building AI responsibly

Mirroring how ISO 27001 governs information security, **ISO/IEC 42001** (published December 2023) is the first certifiable management-system standard for artificial intelligence. It gives an organisation a structured way to govern how AI is developed and used — risk assessment, data governance, transparency, and human oversight — and is quickly becoming the way to *demonstrate* responsible AI to customers and regulators, just as SOC 2 and ISO 27001 do for security.

> **The two AI risks the vibe coder cannot see**
>
> **Building with AI:** an AI coding assistant may transmit your proprietary source or customer data to a third party, and code it generates can carry licensing and IP-provenance questions or subtle vulnerabilities you did not write and cannot see. **Building AI in:** the moment your product makes automated decisions about people — who gets hired, approved, or flagged — you may have crossed into high-risk territory with obligations for data governance, explainability, bias testing, and human oversight that must be designed in, not added after launch.

10. Secure Development and the Software Supply Chain

Every framework in this paper assumes something it rarely spells out for the builder: that software is developed *securely*, and that you know what is inside it. For a fast-moving team, two disciplines carry most of the weight — a secure development lifecycle, and supply-chain awareness — and both are exactly the kind of invisible work an AI assistant will not add unless asked.

A secure development lifecycle

Secure development means weaving security through how code is built rather than testing for it at the end. In practice that is a short, repeatable set of habits: **threat-model** a feature before building it (ask how it could be abused, not just how it should work); check work against a known list of common failures such as the **OWASP Top 10** (injection, broken access control, authentication flaws); require **code review** on changes to sensitive paths; and run automated security scanning in the pipeline so regressions are caught before release. None of this is exotic — it is the difference between hoping code is secure and having a reason to believe it is.

Knowing your supply chain

Modern applications are mostly other people's code — open-source packages, pulled in transitively by the hundred. That is normal and necessary, but it means your security depends on components you did not write. Recent history is a warning: **SolarWinds** (2020) saw attackers compromise a trusted build pipeline to

reach thousands of downstream organisations; **Log4Shell** (2021) turned a single flaw in a ubiquitous logging library into a global emergency; and the **xz-utils** backdoor (2024) was a near-miss in which a malicious maintainer almost planted a backdoor in a core Linux component. The defensive baseline is knowing what you ship — maintaining a **Software Bill of Materials (SBOM)**, scanning dependencies for known vulnerabilities, and updating deliberately rather than blindly.

> **Why this belongs in a compliance paper**
>
> Secure-SDLC and supply-chain controls are not optional extras — they appear, in some form, in ISO 27001 (secure coding, supplier and change management), SOC 2 (the Common Criteria on change management and risk), and increasingly in regulation. A vibe coder who assembles a product from AI-generated code and a hundred unreviewed dependencies has taken on real risk in precisely the area frameworks scrutinise most, and can see least.

11. Why This Is Critical, Not Optional

The cost curve of retrofitting

The single most important economic fact about compliance is that its cost rises steeply the later it is addressed. A decision made at design time — "every table that holds personal data carries a user ID and a deletion hook" — costs minutes. The same decision made after two years of production data, across a dozen services and an analytics pipeline, can be a multi-month re-architecture. Security and privacy controls are foundational; adding a foundation to a building that is already standing is the expensive kind of work. This is why **build it in early** is not a slogan but a cost-avoidance strategy.

The commercial case

For any product selling to businesses, compliance is a revenue gate. Enterprise procurement routinely requires a SOC 2 Type II report or ISO 27001 certificate before a contract can be signed; without one, deals stall in security review or never start. For products touching EU users, GDPR compliance is the price of legally operating in the market at all. Compliance is not a cost centre that competes with growth — increasingly, it is a precondition for it.

Trust, and the cost of getting it wrong

Beyond contracts and fines, these standards exist because the failures they prevent are real and expensive: data breaches, regulatory penalties, and the erosion of customer trust that is far harder to rebuild than any system. The controls are, in effect, institutional memory — each one encodes a lesson someone else learned painfully. Ignoring them is not saving effort; it is choosing to relearn those lessons on your own customers.

The numbers are not hypothetical

Regulators have shown they will impose penalties at a scale that is existential for most companies. The largest fines cluster around exactly the failures this paper warns about — collecting and moving personal data without a lawful basis or adequate safeguards.

Penalty	Who / when	What went wrong
€1.2 billion	Meta, 2023 (largest GDPR fine to date)	Transferring EU users' personal data to US servers without adequate safeguards.

Penalty	Who / when	What went wrong
€746 million	Amazon, 2021 (under appeal)	Advertising and consent practices found non-compliant with the GDPR.
€530 million	TikTok, 2025	Transfers of EU user data to China without adequate protection.
Up to €35M or 7%	EU AI Act ceiling (per breach)	Reserved for prohibited AI practices — higher than the GDPR's own ceiling.

> **The lesson in the fines**
>
> These are not obscure edge cases — they are the biggest, best-resourced technology companies in the world, penalised for mishandling ordinary personal data such as identifiers and IP addresses. If organisations with entire legal and security departments get this wrong at this scale, an early-stage team building on intuition is not somehow safer; it is simply not yet large enough to have been noticed.

12. Compliance Maturity by Stage

One of the most common mistakes is applying the wrong level of rigour for the company's stage — either burning scarce runway chasing certifications and reports a five-person team does not yet need, or, far more often, building a foundation that quietly makes them prohibitively expensive to reach later. The aim is not to “do everything now.” It is to do the cheap, foundational things early and add formal, evidenced compliance exactly when the business needs to unlock it.

Stage	What's realistically expected	Why it matters here
Pre-seed / MVP	Hygiene, not certificates: secrets in a managed store, HTTPS, basic access control, collect only data you need, a real privacy policy.	No auditor is watching yet — but the schema and architecture you set now decide how painful compliance is later.
Seed	Data classification, audit logging, least-privilege access, a documented data lifecycle, GDPR basics if you have EU users. Start keeping evidence.	First security questionnaires and investor questions arrive; early enterprise pilots begin to probe.
Series A	SOC 2 Type II underway or achieved; ISO 27001 if selling to global/EU enterprise; records of processing and a DPO if GDPR-heavy.	Compliance becomes a revenue gate and diligence intensifies; the absence of a report now costs deals.
Growth / B+	Multiple frameworks, a dedicated security/compliance owner, vendor-risk program, continuous monitoring.	Larger attack surface, real regulatory exposure, and enterprise procurement that assumes maturity.

> **Right-sizing the investment**
>
> A pre-seed team does not need SOC 2 — but it does need to avoid building the one-way doors that make SOC 2 a re-architecture later. Under-investing looks like collecting everything, hard-coding secrets, and giving everyone production access. Over-investing looks like a two-person team spending six months and six figures on a certificate no customer has yet asked for. Match the effort to the stage; keep the foundations cheap and correct at every stage.

13. The Cost and Timeline Reality

Founders consistently underestimate that these are *elapsed-time* problems as much as budget problems. A SOC 2 Type II report in particular cannot be bought quickly at any price, because it attests to controls operating over a window of time that must actually pass. The figures below are broad industry ranges for a startup or mid-sized company; actual cost varies widely with scope, headcount, and how much you build in advance versus retrofit.

Framework	Typical time to first result	Typical all-in cost	Notes
SOC 2 Type II	~6–12 months (incl. a 3–6 month observation window)	~$25k–$150k+ all-in (audit alone often $7k–$100k)	Cannot be rushed; the observation period must genuinely elapse with controls running.
ISO 27001	~4–8 months to first certification (faster with automation)	~$50k–$200k (materially less for small startups on platforms)	Valid 3 years with annual surveillance audits; certifies the organisation, not one product.
GDPR	Ongoing — no certificate exists	Mostly engineering + legal/DPO time	Not a project with an end date; a continuous operating obligation.

Compliance-automation platforms (such as Vanta, Drata, Secureframe and Sprinto) have compressed these timelines meaningfully by automating evidence collection, control monitoring, and questionnaire responses. They are genuinely useful — but it is important to understand their limit: **they automate the proof, not the substance**. A platform will tell you a control is failing; it will not build the control, write your access model, or refactor a system that cannot delete a user. The engineering still has to exist.

> **The cost everyone forgets**
>
> The audit fee is the visible number. The larger, hidden cost is engineering rework — re-architecting data models, adding audit logging after the fact, untangling access — for teams that did not build it in. That retrofit routinely dwarfs the audit itself, which is the entire economic argument for treating these standards as design inputs from day one.

14. The Founder's Validation Playbook

Most founders learn the state of their compliance posture from an auditor or an acquirer — the two most expensive people to hear it from. The alternative is to validate it yourself, early, and the key discipline is simple: **do not accept a claim that a control exists — prove it operates**. “We log access” and “we can delete a user” are hypotheses until someone actually produces the log and completes the deletion. The tests below are things a founder can run, or ask an engineer to run, in an afternoon.

The claim	How to actually test it (don't just ask)
“Access is least-privilege”	Pull the current production access list. Count who has standing access and justify each one. Surprises here are the norm, not the exception.
“We log security events”	Ask an engineer to show who accessed a specific customer's record last month. If they can't produce it, you are not logging it.

The claim	How to actually test it (don't just ask)
“We can delete a user”	Pick a test account and delete it end-to-end — app, database, backups, analytics, third parties. Time how long it takes and what's left behind.
“Secrets are secure”	Search the repository's full history for keys and passwords. Confirm no plaintext credentials live in source, config, or CI.
“Data is encrypted”	Confirm TLS on every endpoint and at-rest encryption on every datastore — including backups and that one legacy database everyone forgot.
“We know our vendors”	List every sub-processor that touches customer data and confirm a data-processing agreement exists for each. Fourth-party risk is still your risk.

A four-step self-validation loop

1. **Map your data.** Write down every category of personal and sensitive data, where it lives, who can reach it, and how long you keep it. You cannot protect or prove what you have not inventoried.
2. **Run a gap analysis.** Compare that map and your controls against your target framework — using a checklist or an automation platform — and list what's missing.
3. **Test, don't assume.** Run the six tests above. A control that has never been exercised is a control you don't actually have.
4. **Gather evidence continuously.** Keep dated records of decisions, access reviews, and incidents as you go. Retroactive evidence is the single hardest thing to produce in any audit.

> **Founder red flags — signs a retrofit is coming**
>
> Nobody can say exactly what personal data you hold · every engineer has production database access · there is no audit log of who saw what · “delete my account” doesn't really delete the data · secrets live in the codebase · you've never listed your sub-processors · security is “something we'll do before the raise.” Any two of these means the bill is accruing quietly.

15. The Investor Due-Diligence Question Bank

For an investor, security and privacy posture is not a compliance checkbox — it is **unpriced risk and a revenue predictor**. A company that cannot sell to enterprises without a SOC 2 report has a capped near-term market; a company holding EU personal data with no deletion capability carries a contingent liability that surfaces at the worst possible moment, often during a later acquisition's diligence. The questions below, and crucially *why* and *where* each one bites, let an investor price that risk before the term sheet rather than after.

Question to ask	Why it matters	Where / when it's critical
What personal or regulated data do you hold, and what's your lawful basis for it?	Sizes GDPR and breach exposure; uncapped data liability can survive into an acquisition.	Any consumer or EU-facing product; sharpens at Series A+.
Do you have SOC 2 or ISO 27001 — or a dated, credible plan to get there?	Determines whether enterprise revenue is even reachable in the near term.	B2B/SaaS, before Series A and B raises.
Can you completely delete a	Tests whether GDPR rights are real	EU users or any privacy-sensitive

Question to ask	Why it matters	Where / when it's critical
user on request, including backups?	and exposes hidden architectural debt.	product.
Show me your access controls and audit logs.	Reveals insider-risk exposure and whether a breach could even be reconstructed.	Fintech, health, or anything holding sensitive data.
Have you had any breaches, incidents, or contact from a regulator?	Undisclosed incidents are a classic late-diligence deal-breaker.	Always — and re-confirm at every round.
Who owns security, and what is the budget?	Distinguishes a resourced program from an aspirational slide.	Series A onward, as headcount and data grow.
Is compliance built into the product, or bolted on later?	Predicts future rework cost and technical debt — the core thesis of this paper.	Always; it is the cheapest question with the most predictive power.

> **For investors: watch the answer behind the answer**
>
> Often the specific answer matters less than whether the founder can answer at all. A founder who can immediately say what data they hold, who can access it, and how they'd delete it is describing a controlled system. A founder who hesitates on those basics is describing risk they haven't priced — and that gap, more than any missing certificate, is the signal worth weighing.

16. Closing the Gap: A Practical Path

The goal is not to slow builders down or bury them in process, but to make the invisible constraints visible early and encode them into defaults so that the fast path is also the compliant path. The following moves deliver most of the value.

5. **Classify data on day one.** Know what personal, sensitive, and regulated data your product touches before you build the schema. Everything else follows from this.
6. **Make least privilege the default.** New services, roles and people start with no access and are granted only what they need. Nobody gets standing production access by default.
7. **Log consequential actions from the start.** Immutable audit trails cannot be created retroactively, and both SOC 2 Type II and GDPR breach-response depend on them.
8. **Design the data lifecycle, including deletion.** Every category of personal data needs a lawful basis, a retention period, and an automated deletion path — the right to erasure is an architecture decision.
9. **Manage secrets properly.** A managed secret store from the first commit; never credentials in source control.
10. **Encrypt in transit and at rest by default.** This is table stakes across every framework and trivial to set up early.
11. **Treat the AI assistant as a junior engineer.** Prompt explicitly for the invisible requirements — access control, validation, logging, error handling — and review its output against them. It will not add them unasked.

12. **Write things down as you go.** Every standard demands evidence. A short, current record of decisions and controls turns a future audit from an archaeology project into a formality.

> **The mindset shift**
>
> The vibe coder asks “does it work?” The compliant builder also asks “who can see this data, how long do we keep it, how would I prove that, and how would I delete it?” Those four questions, asked at design time, close most of the gap — and none of them require slowing down, only remembering to ask.

17. If You're Already Behind: A Remediation Playbook

Most of this paper argues for building compliance in from the start — but many teams reading it will already have shipped, grown, and only now discovered the gap. That is the common case, not the shameful exception, and it is recoverable. The goal is not to stop everything and boil the ocean; it is to **triage** — stop the bleeding, then close gaps in order of risk. The sequence below is how to dig out without halting the business.

13. **Stop making it worse.** Immediately fix the cheap, high-impact hygiene items: rotate and vault any secrets in source control, remove standing production access nobody needs, and turn on encryption and logging where they're missing. These are hours of work that cap your exposure today.
14. **Inventory before you architect.** You cannot fix what you cannot see. Map what data you hold, where it lives, who can reach it, and which third parties touch it. Almost every later step depends on this map.
15. **Triage by risk, not by framework.** Rank gaps by likelihood and blast radius — a database of personal data with no access controls outranks a missing policy document. Fix the things an attacker or a regulator would reach first.
16. **Retrofit the irreversible things first.** Audit logging and user-deletion capability can't be created retroactively and gate both SOC 2 Type II and GDPR. Start them early so the observation clock and the evidence trail begin as soon as possible.
17. **Adopt tooling to buy back time.** A compliance-automation platform will map your current state to a framework, surface gaps, and collect evidence continuously — compressing months of manual work, provided you still do the underlying engineering.
18. **Sequence the frameworks to revenue.** Pursue the framework that unblocks the nearest deal or market first (often a SOC 2 report for US enterprise, GDPR for EU users), rather than attempting everything at once.

> **The reframe**
>
> Being behind is a debt, not a verdict. Like technical debt, it compounds if ignored and is entirely payable if you service it deliberately. The worst outcome is not discovering the gap — it is discovering it and continuing to build new one-way doors on top of it. Draw the line today, then pay it down in risk order.

18. Conclusion

The tools that let a single builder ship a complete product in a day are among the most empowering developments in software's history. They are not the problem. The problem is that velocity, uncoupled from the context that once travelled with slower development, makes it effortless to build something that

works perfectly and complies with nothing. ISO 27001, SOC 2 and the GDPR are not obstacles to that velocity — they are the encoded difference between a product that demos well and one that an enterprise, a regulator, and a customer can actually trust.

The vibe-coder gap is real, but it is not a failure of skill or care — it is a failure of *visibility*. The constraints are invisible until an audit, a questionnaire, or a breach makes them violently visible, and by then they are expensive. The answer is to surface them at the only point where they are cheap: the design. Classify the data, default to least privilege, log from the start, plan for deletion, and ask the four questions. Do that, and compliance stops being the thing that slows you down after you have built the wrong foundation — and becomes simply the way you build.

Appendix A: Glossary of Terms

A quick reference to the acronyms and terms used in this paper, for readers coming from different backgrounds.

Term	Meaning
ISO/IEC 27001	International standard for an Information Security Management System (ISMS); certifiable. Current edition: 2022.
ISMS	Information Security Management System — the processes an organisation uses to manage security risk, at the heart of ISO 27001.
Annex A	The reference set of 93 security controls in ISO 27001:2022, grouped into organizational, people, physical and technological themes.
SOC 2	An AICPA attestation report on a service organisation's controls against the Trust Services Criteria.
TSC	Trust Services Criteria — the five SOC 2 categories: Security, Availability, Processing Integrity, Confidentiality, Privacy.
Type I / Type II	SOC 2 report variants: Type I tests control design at a point in time; Type II tests operating effectiveness over a period (3–12 months).
GDPR	EU General Data Protection Regulation — the law governing processing of EU residents' personal data.
DPA	Data Processing Agreement — a contract governing how a vendor (processor) handles personal data on your behalf.
DPIA	Data Protection Impact Assessment — a risk assessment required before higher-risk personal-data processing.
DPO	Data Protection Officer — a designated privacy-compliance role required for certain organisations under the GDPR.
Sub-processor	A third party that processes personal data on behalf of your organisation (e.g. a cloud or analytics vendor).
PII	Personally Identifiable Information — data that identifies an individual, such as name, email, or IP address.
Least privilege	The principle of granting each user or service only the minimum access needed to do its job.
Audit log	An immutable record of security-relevant actions (who did what, when) — required evidence in most frameworks.
EU AI Act	The EU's risk-based law regulating AI systems, phased in from 2024 to 2027, with fines up to €35M or 7% of turnover.
GPAI	General-Purpose AI — foundation/large models subject to specific EU AI Act obligations from August 2025.
ISO/IEC 42001	The first certifiable management-system standard for artificial intelligence (published 2023).
OWASP Top 10	A widely used list of the most critical web-application security risks.
SBOM	Software Bill of Materials — an inventory of the components and dependencies that

Term	Meaning
	make up your software.
PCI DSS / HIPAA / NIST	Standards for payment-card data, US health information, and US federal cybersecurity, respectively.

Important disclaimer — please read

This document is provided for general information and educational purposes only. It is **not legal, regulatory, or professional compliance advice**, and it does not create any advisory relationship. Security, privacy and compliance regulations **change frequently and vary significantly by jurisdiction** — the standards, control counts, dates, penalty amounts, costs and timelines cited here reflect the author's understanding at the time of publication and may since have changed. Requirements that apply to your product depend on your specific data, customers, sector and locations. **Always verify the current requirements for your jurisdiction against official sources and qualified legal and security professionals before making decisions.** The author accepts no liability for actions taken on the basis of this document.

Last updated: Monday, 13 July 2026, 09:48 (ICT, UTC+7). Verify against current sources before relying on any figure or date herein.

About the author

Justin James — writing on building secure, compliant products. More at justinjames.agencie.io.

Sources & further reading

ISO/IEC 27001:2022 and ISO/IEC 42001:2023 (International Organization for Standardization); AICPA SOC 2 Trust Services Criteria; Regulation (EU) 2016/679 (GDPR), Articles 5, 25, 33; Regulation (EU) 2024/1689 (EU AI Act); PCI DSS (PCI SSC); HIPAA Security Rule; NIST Cybersecurity Framework and SP 800-53; OWASP Top 10. ISO control counts, the 31 October 2025 ISO transition deadline, the EU AI Act phased-application dates and penalty ceilings, and the referenced GDPR fines (Meta €1.2bn 2023; Amazon €746m 2021; TikTok €530m 2025) were verified against current published guidance. Cost and timeline figures are broad industry ranges (2025–26) and vary significantly with scope, company size, and readiness; treat them as planning estimates, not quotes.