



Professional Services, Software & Solutions Group

WHITE PAPER

AI Transformation: From Pilots to Production

A practical white paper on turning generative AI, agents and machine learning into working systems that an organization of any size can run, govern and afford.

PS3G Inc.

1201 N Market Street, Suite 111, Wilmington, DE 19801 · Email hello@ps3g.com

Contents

- **01** Why this wave is different from the last one
- **02** The four AI patterns that actually ship
- **03** Build, buy or integrate
- **04** The AI transformation roadmap
- **05** Data readiness and data governance for AI
- **06** Reference architecture
- **07** Security, privacy and compliance
- **08** Economics: what AI really costs and how to budget it
- **09** A 90-day plan, a readiness checklist and three illustrative examples

Who this is for

This document is written for the people who decide whether AI gets money and attention in their organization: owners and founders of small businesses, department heads and operations leaders in mid-market companies, and IT and digital leaders in enterprise divisions. It assumes no technical background. Where a technical term is unavoidable, it is explained the first time it appears.

It is deliberately neutral about vendors, clouds and models. The patterns, decision tables and checklists apply whether you run on Azure, AWS or Google Cloud, use a commercial model API or an open-weight model, and have two employees or two thousand.

How to use this document

Read sections 01 to 04 in order; they set the vocabulary and the overall approach. Sections 05 to 08 are reference material: come back to them when planning data work, architecture, security review or budget. Section 09 is the part to copy into your own planning documents. The three illustrative examples at the end are hypothetical and show what a sensible first project looks like at different scales; they do not recommend a particular use case.

A note on numbers. This paper contains no survey statistics or market forecasts. Where we give a figure, it is a rule of thumb from the PS3G team's own delivery experience and is labelled as such. Use it as a starting point for your own estimates, not as a benchmark.

01 Why this wave is different from the last one

Most organizations have been through an "automation" push before. Robotic process automation, predictive models and chatbots promised a lot in an earlier wave and delivered unevenly. It is fair to ask what is actually different this time, and what is not.

What changed

That earlier wave rested on two technologies. RPA scripts clicked through screens the way a person would, and broke the moment a screen changed. Classical machine learning models were trained from scratch on your own labelled

data, which meant months of preparation before anything useful appeared, and a data-science team to keep it running.

Large language models (LLMs) change the economics of both. A general-purpose model, reached through an API, can read a contract, summarize a support ticket, draft a reply, extract fields from a scanned form or write a database query without being trained on your data first. That removes the biggest cost of the previous wave. Three further shifts matter now:

- **Copilots are embedded in tools you already pay for.** Office suites, CRMs, help desks and developer tools now ship with an assistant. Some value is available by switching on a feature, not by building anything.
- **Agents can take actions, not just produce text.** A model can call your systems, check a result and decide what to do next. This is where the biggest upside and the biggest risk both sit.
- **Regulation has arrived.** The EU AI Act, the Colorado AI Act (SB 24-205, now taking effect) and frameworks such as NIST AI RMF 1.0 and ISO/IEC 42001 mean "we are experimenting" is no longer a complete answer to a board or to a customer's security questionnaire.

What did not change

The reasons projects stalled in the last wave are still the reasons AI projects stall. In PS3G's experience the pattern is consistent across company sizes:

Failure mode	First wave	Today
The data was not ready	No labelled training set; fields inconsistent across systems	Documents scattered across drives and inboxes; no owner for the source of truth
No one owned the process	Bot automated a process nobody had documented	Assistant answers questions nobody has agreed the correct answers to
Pilot never became a product	Model lived in a data scientist's notebook	Prototype lives in one person's chat window; no monitoring, no fallback, no budget line
Nobody measured it	"Hours saved" estimated once, never verified	Same, plus unmanaged token spend

The practical lesson. The technology is far more capable, but the discipline required to get value from it is unchanged: a named process owner, a measurable outcome, clean enough data, a path from pilot to production, and a budget that survives the demo. This paper is about that discipline.

02 The four AI patterns that actually ship

Strip away the product names and almost every AI system that reaches production today is one of four patterns, or a combination of them. Naming the pattern early makes scoping, pricing and risk review far simpler.

1. Assistants and copilots

A person asks, the model answers, the person decides. Drafting, summarizing, explaining, coding. A human reviews every output, which keeps risk low and lets you start the same week.

2. Retrieval-augmented knowledge systems (RAG)

The model answers from *your* documents: policies, contracts, manuals, tickets. Relevant passages are retrieved for each question and the answer cites them. Answer quality is capped by document quality.

3. Workflow automation with AI steps

A defined process (intake, triage, classification, extraction, routing) where some steps are performed by a model and the rest by ordinary software, with a confidence threshold and a human exception queue. Most measurable savings come from here.

4. Autonomous agents with guardrails

The model plans a sequence of actions, calls tools, checks results and iterates toward a goal. Appropriate only where the action space is bounded, every action is logged and a human can approve or roll back.

Decision table: which pattern fits

Pattern	Fits when	Main risks	Data needed	First result
Assistant / copilot	Staff spend time writing, reading or searching; outputs are reviewed before use	Leakage to consumer tools; over-trust in fluent but wrong text	None beyond the prompt; optional style guides and templates	Days to weeks
RAG knowledge system	Staff or customers repeatedly ask questions whose answers exist in documents	Stale or conflicting documents; answers without citations; access-control gaps	A curated, owned, access-labelled document set	Four to eight weeks
Workflow with AI steps	A high-volume, repetitive process with clear inputs and a definable correct output	Silent errors at scale; no exception handling; undocumented process	Historical examples with known correct outcomes for testing	Six to twelve weeks
Autonomous agent	Multi-step tasks across systems where each step is reversible or approvable	Unintended actions; prompt injection via tool outputs; runaway cost	Clean APIs, audit logging, test environments	A quarter or more, after a workflow is already live

Order matters. The PS3G team's rule of thumb: ship pattern 1 or 2 first, then 3, and only then 4. Organizations that start with an autonomous agent usually discover that the data, APIs and governance the agent needs were the real project all along.

03 Build, buy or integrate

For any use case there are three ways to get it: switch on a copilot inside software you already license, call a model API from your own application, or build a custom system around a model. Most

organizations will do all three. The mistake is treating one of them as the default for everything.

Option	What it is	Choose it when	Watch out for
Buy: SaaS copilot	An AI feature inside a product you already use (office suite, CRM, help desk, ERP, developer tools), priced per user	The use case is generic (drafting, meeting notes, search within that product) and the data already lives there	Per-seat cost across the whole company; little control over what the vendor processes; vendor terms on training and retention
Integrate: model API	Your own application calls a hosted model (commercial or open-weight) through a cloud API, with your prompts, data and rules	The use case is specific to your process, needs your data, or must sit inside a system you own	Needs an engineering owner; prompt and model changes must be tested; usage must be metered and capped
Build: custom system	A full application: retrieval, orchestration, evaluation, monitoring, fine-tuning where justified, often several models	The capability is a differentiator, volume is high, or compliance requires full control of the data path	Longest time to value; ongoing operations; easy to over-build when a simpler option would do

A quick scoring method

Score each candidate use case from 1 (low) to 3 (high) on the four questions below. In PS3G's experience totals of 4 to 6 usually point to "buy", 7 to 9 to "integrate" and 10 to 12 to "build". The scores start a conversation; they are not a verdict.

1. **Specificity.** How much does the task depend on our own process, vocabulary and data rather than general knowledge?
2. **Data sensitivity.** How much would we lose if this data were processed or retained by a third party on their terms?
3. **Volume.** How many times a day will this run? Per-seat pricing gets expensive across many users; per-token pricing gets expensive at very high volume.
4. **Differentiation.** Would a competitor doing this exactly as well as us hurt our position?

A small business usually starts with "buy" for writing and meetings, then one "integrate" project wired into the quoting or intake system it already runs on. A mid-market company or enterprise department typically runs a licensed copilot, a few "integrate" projects owned by specific teams, and one "build" where the capability is strategic. The governance work in section 07 is the same for all three.

Vendor-neutral by design. Whatever you choose, keep the model behind an interface you control. Models improve and reprice every few months; a system that can swap its model in a day is worth far more than one locked to a single provider.

04 The AI transformation roadmap

"Transformation" sounds like a multi-year program. In practice it is a repeating loop of five stages, run first on one use case and then on the next. The loop is the same for a three-person firm and a three-thousand-person division; only the number of parallel loops differs.

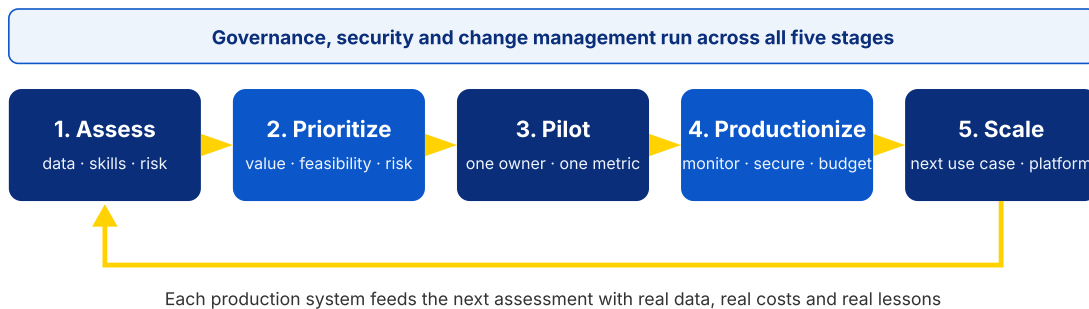


Figure 1 – The five-stage AI transformation loop. Governance is a lane that runs alongside, not a gate at the end.

- 1 **Assess (one to three weeks).** Inventory where documents and data live, who owns them, which systems have usable APIs, what AI tools staff already use (approved or not) and which regulations apply. Output: a one-page readiness summary and a list of about ten candidate use cases.
- 2 **Prioritize.** Score each candidate on value (hours, revenue, risk reduced), feasibility (data ready, pattern proven, owner available) and risk (data sensitivity, customer impact, regulatory exposure). Pick one with high value, high feasibility and low risk, not the most impressive one.
- 3 **Pilot (four to eight weeks).** Build the thinnest version a real user can try on real work. Define the success metric before the first line of code and measure against a baseline. A pilot that cannot fail is not a pilot.
- 4 **Productionize.** Add what the demo lacked: authentication, logging, monitoring, evaluation tests, a fallback when the model is down, cost caps, a support owner and a budget line. This stage is usually as long as the pilot and is the one most often skipped.
- 5 **Scale.** Reuse the retrieval layer, the evaluation harness and the governance paperwork for the next use case. The second project should cost noticeably less than the first; if it does not, something was not built as a platform.

05 Data readiness and data governance for AI

Generative AI removed the need to train models on your data. It did not remove the need for your data to be findable, correct, current and permissioned. In PS3G's experience the data work is where most of the calendar time in an AI project goes, in organizations of every size.

The usual blockers

Blocker	What it looks like	What to do
No source of truth	Three versions of the price list, the policy manual or the product spec, in three places	Name an owner per document class; pick the authoritative location; archive or label the rest
Documents without metadata	No owner, date, status or sensitivity label, so a system cannot tell draft from final or public from internal	Add a minimal metadata set (owner, date, status, audience, sensitivity) at folder or library level before indexing
Permissions are implicit	Access is controlled by who knows the link; an assistant would surface HR files to everyone	Index only what has explicit access controls; the AI layer must inherit them, never bypass them
Structured data locked in screens	Key records exist only in a legacy application with no API or export	Build a read-only data access layer or a scheduled export to a data platform first
Nobody can say what "correct" is	No examples of a good answer, a correctly classified ticket or a properly extracted invoice	Collect fifty to a few hundred real examples with agreed correct outputs; this becomes the evaluation set
Regulated data mixed in	PHI, payment data or personal data in the same folders as general content	Classify, then exclude, redact or route to a compliant processing path

A minimum data governance set for AI

You do not need a full enterprise data governance program to start. You do need the following five things written down for each data set an AI system touches:

- ☐ **Owner** – a named role accountable for accuracy and for approving its use in AI systems.
- ☐ **Classification** – public, internal, confidential or regulated (and which regulation: HIPAA, payment, personal data).
- ☐ **Allowed uses** – which patterns from section 02 may use it, and whether it may leave your cloud tenancy.
- ☐ **Refresh rule** – how often it is re-indexed and who is notified when it changes.
- ☐ **Retention and deletion** – including deletion from indexes, caches and logs, not just the source.

Where the data platform fits. Organizations with a modern data platform (a governed lakehouse or warehouse with lineage and access control) move faster because the owner, classification and access questions are already answered. Those without one can build the minimum set above for one use case and grow it. Either route works; the difference is how much is reusable for project two.

06 Reference architecture

Every production AI system PS3G has delivered, regardless of cloud or model, ends up with the same seven building blocks. Knowing them lets you ask a vendor or a team the right questions, even if you never build one yourself.

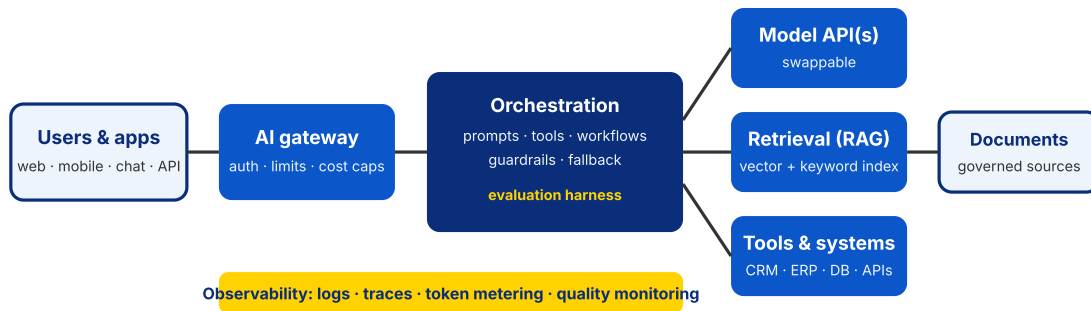


Figure 2 – Reference architecture for a production AI system. The model is one replaceable component, not the center.

Building block	What it does	Question to ask
AI gateway	Single entry point: authentication, per-user and per-app rate limits, spend caps, routing between models, central logging	"Can we cut off one application's spend without touching the others?"
Orchestration	Holds prompts, tool definitions, workflow logic, guardrails (input and output checks) and the fallback when a model fails	"Where do prompts live, and are they versioned and tested like code?"
Model API(s)	Commercial or open-weight models, hosted by a cloud provider or in your tenancy, behind an interface you own	"How long would it take to switch models?"
Retrieval and vector store	Indexes documents as embeddings plus keywords; returns relevant passages with their access labels	"Does retrieval enforce the same permissions as the source system?"
Tools and systems	APIs the model may call: read-only first, write actions behind approval	"What is the list of actions the model may take?"
Evaluation harness	A fixed set of real examples with known good outputs, run on every prompt or model change	"How do we know the last change did not make answers worse?"
Observability and cost controls	Traces per request, token metering by user and feature, quality sampling, alerts on cost and error spikes	"What did this feature cost yesterday, and who used it?"

07 Security, privacy and compliance

AI systems inherit every security concern of ordinary software and add a few of their own. The new ones are manageable, but they must be named explicitly in the design and in the vendor contract, because standard application security reviews do not cover them.

The AI-specific threats

Threat	What it is	Primary controls
Prompt injection	Instructions hidden in content the model reads (an email, a web page, a document, a tool result) that hijack its behavior	Treat retrieved and tool-returned content as untrusted data; restrict tools; require human approval for consequential actions
Data leakage	Confidential or regulated data sent to a model or vendor without authorization, or shown to a user who should not see it	Classify data (section 05); route regulated data only to approved endpoints; permission-aware retrieval; block unapproved consumer tools
Insecure output handling	Model output executed or rendered without checks (generated SQL, code, HTML, links)	Validate and sandbox outputs; never run generated code or queries with elevated rights
Excessive agency	An agent with more tools, permissions or autonomy than the task needs	Least-privilege tool sets; read-only by default; action limits; logging and rollback
Denial of wallet	Abuse or bugs driving unbounded token spend	Per-user and per-app caps at the gateway; anomaly alerts; hard budget stops

These map directly to the OWASP Top 10 for LLM Applications, the reference PS3G uses for AI threat modelling and one your security team or auditor will recognize.

Regulated data: HIPAA and PHI

If protected health information can reach the model, every vendor in the path needs a signed Business Associate Agreement, data must stay in a compliant region and tenancy, prompts and outputs containing PHI fall under the same retention rules as the record system, and vendors must commit in writing not to train on your data. PS3G applies this discipline in its own HIPAA-compliant product work.

Vendor terms to read before signing

- Is customer data used for training? (It should not be.)
- Retention of prompts, outputs and logs, and the deletion process.
- Data residency and the sub-processor list.
- Security attestations (SOC 2 Type II or equivalent) and incident notification terms.
- Price-change and model-deprecation notice periods.

Governance frameworks. For a program that will stand up to a board, a customer or a regulator, map your controls to NIST AI RMF 1.0 (govern, map, measure, manage) and, where certification is wanted, ISO/IEC 42001. If you serve EU customers or make consequential decisions about Colorado residents (employment, lending, housing, healthcare, insurance), review the EU AI Act and Colorado SB 24-205 early; both require documentation that is far easier to produce during design than after launch.

08 Economics: what AI really costs and how to budget it

The smallest line item in an AI project, the model itself, gets the most attention. In PS3G's experience the model API is rarely the largest cost of a production system in its first year. Knowing the real cost drivers prevents both over-spending on the wrong thing and abandoning a good project because the pilot's token bill looked alarming.

Cost drivers, in rough order of size for a first project

1. **People time.** Engineering, data preparation and subject-matter review. The experts who build the evaluation set and review outputs are the most underestimated line.
2. **Integration and data access.** Connecting to the systems that hold the data, cleaning it, and building the permission-aware index.
3. **Platform and run costs.** Cloud compute, vector store, gateway, logging and monitoring: ordinary cloud costs that scale with usage.
4. **Licenses.** Per-seat copilot fees: small per person, large across a company.
5. **Model usage (tokens).** Charged per unit of text in and out. Variable, easy to cap, and usually the cheapest line until volumes are very high.

Token and usage budgeting: rules of thumb

- **Measure in the pilot, then multiply.** Average tokens per request times expected requests per day gives a monthly estimate; add a margin for retries and long documents.
- **Context is the cost.** Retrieved documents and conversation history usually dwarf the question and answer. Tight retrieval and history trimming are the biggest levers.
- **Use the smallest model that passes the evaluation.** Route easy requests to a cheaper model and reserve the strongest for hard cases; this is a gateway rule, not a rewrite.
- **Cap first, optimize later.** A hard monthly cap per application protects the budget while the real pattern of use emerges.

Sizing a first project and choosing an engagement model

A first project should be small enough to finish inside one quarter, large enough that its result changes a decision, and bounded enough that its full cost fits on one page. If the business case needs a multi-year model to look good, choose a different first project.

Engagement model	Best for	How it is priced	What you commit to
Fixed scope	A well-defined pilot or a single production use case with a known pattern	Fixed price for a written scope, milestones and acceptance criteria	Timely access to data, systems and a decision-maker
Dedicated team	A roadmap of several use cases; building an internal AI capability over time	Monthly rate per named role; flexes quarter to quarter; round-the-clock delivery possible	A product owner on your side and a backlog
Managed service	Running production AI systems: monitoring, evaluation, model updates, cost management, compliance evidence	Monthly fee tied to systems covered and service levels	Agreed runbooks, escalation paths and change windows

09 A 90-day plan, a readiness checklist and three illustrative examples

Ninety days is long enough to take one use case from idea to a monitored production system, and short enough that the organization stays interested. The plan below is PS3G's starting point; adjust the

weeks to your own pace.

The 90-day plan

- 1 Days 1–15: Assess and choose.** Run the readiness checklist. Hold one workshop with the people who do the work, not only those who manage it. Pick one use case, one owner, one metric and one baseline measurement.
- 2 Days 16–30: Prepare the ground.** Collect the evaluation set of real examples with agreed correct outputs. Confirm classification and access rules for every source. Settle vendor terms and the staff acceptable-use policy. Stand up the gateway with cost caps, even for the pilot.
- 3 Days 31–60: Pilot with real users.** Build the thinnest working version and put it in front of five to ten real users on real work. Measure against the baseline weekly. Run the evaluation set on every change. Record every failure; the failures define the production backlog.
- 4 Days 61–85: Productionize.** Authentication, permission-aware retrieval, logging and traces, monitoring and alerts, fallback behavior, a security review against the OWASP LLM Top 10, the governance file, a support owner and a budget line.
- 5 Days 86–90: Decide and plan the next loop.** Present the measured result, the full cost and the lessons. Decide: scale, keep as is, or stop. Choose the next use case using what the platform now makes cheap.

One-page AI readiness checklist

Business and people

- ☐ Named executive sponsor and named process owner
- ☐ One use case with a written metric and a baseline
- ☐ Subject-matter experts have review time allocated
- ☐ Staff acceptable-use policy for AI tools communicated

Data and systems

- ☐ Sources identified, owned and classified
- ☐ Access controls explicit and inheritable by an index
- ☐ Evaluation set of real examples collected
- ☐ Regulated data (PHI, payment, personal) routed or excluded

Security and governance

- ☐ Vendor terms reviewed: no training, retention, residency, BAA if needed
- ☐ Threat model covers injection, leakage and excessive agency
- ☐ Controls mapped to NIST AI RMF 1.0 (or ISO/IEC 42001)
- ☐ Applicable law identified (EU AI Act, Colorado SB 24-205, HIPAA)

Platform and money

- ☐ Gateway with authentication, rate limits and hard cost caps
- ☐ Logging, tracing and token metering per user and feature
- ☐ Model behind an interface you control and can swap
- ☐ Full first-year cost on one page; engagement model chosen

Three illustrative examples

The examples below are hypothetical and simplified. They show the same loop, patterns and controls at three very different scales, with a deliberately modest first project at each.

A small business: a regional service company with twelve staff

Situation. Two office staff spend much of each day answering the same scheduling and pricing questions and preparing quotes from a template. **Pattern.** A drafting copilot (buy) plus a small RAG assistant (integrate) that answers staff questions from the company's own service manual, price list and policies, with citations. **Data work.** One afternoon to pick the authoritative versions of about forty documents and label them internal or public. **Controls.** Business-tier license with no-training terms; a gateway cost cap; a staff rule that customer personal data never goes into consumer tools. **Metric.** Time from inquiry to quote sent, measured two weeks before and after. **Lesson.** Document hygiene, not technology, was the limit.

A mid-market company: a distributor with a few hundred employees

Situation. Customer orders arrive as emails and PDFs in a dozen formats and are keyed by hand into the ERP, with frequent errors. **Pattern.** Workflow automation with AI steps: a model extracts line items, quantities and references; a rules layer validates them against the catalog and customer records; anything below a confidence threshold goes to a human queue inside the existing ERP. **Data work.** Three hundred historical orders with their correct ERP entries become the evaluation set; the ERP gets a read-only API for catalog lookups. **Controls.** The model never writes to the ERP directly; a staff member approves each batch for the first month, then only exceptions. **Metric.** Keying errors per hundred orders and hours of order entry per week. **Lesson.** The threshold and the exception queue, not the model, made the system trustworthy.

An enterprise department: the claims operations unit of a large insurer

Situation. Several hundred adjusters search policy documents, correspondence and guidelines for every file. The corporate copilot cannot see the claims systems. **Pattern.** A permission-aware RAG knowledge system in the claims workstation, built on the enterprise data platform, with a later roadmap toward an agent that assembles a file summary. **Data work.** The data platform already supplied ownership, classification and lineage; the project added document-level access labels and a refresh pipeline. **Controls.** Retrieval inherits the claims system's entitlements; prompts and outputs are logged under records retention; the model runs in the company's cloud tenancy under enterprise terms; controls are documented against NIST AI RMF 1.0 because the decisions fall under state AI regulation. **Metric.** Minutes to locate the governing policy clause, sampled weekly. **Lesson.** The governance file took longer than the build, and was why the second use case was approved in weeks.

What the three have in common. One owner, one metric, a bounded use case, an evaluation set of real examples, a human exception path, a cost cap and vendor terms read before signing. None started with an autonomous agent. All three left behind a platform that made the next project cheaper.



Professional Services, Software & Solutions Group

About PS3G

PS3G Inc. (Professional Services, Software & Solutions Group) is an IT services and software company headquartered in Wilmington, Delaware, with an India delivery center that allows round-the-clock engineering, testing and support. We work with organizations of every size, from owner-run businesses to enterprise departments, and we design engagements to fit the size of the problem rather than the size of the client.

Our services include websites and web applications, mobile apps, .NET and Microsoft engineering, cloud migration and DevOps, data platforms and data governance, AI consulting and development, AI governance, cybersecurity and compliance, software testing, managed services, and IT staffing and dedicated teams.

We also build and run our own products. DocVilla is a HIPAA-compliant cloud EHR, telehealth and practice-management platform used by medical practices across the United States. Running a regulated product of our own is why the data governance, security and operational discipline described in this paper are things we practice, not only things we advise.

How we work

Every engagement starts the same way: email us, we reply with questions, then a 30-minute video call that we schedule, a short assessment, and a written proposal with scope, timeline and price. From there we work in whichever model fits: a fixed-scope project, a dedicated team, a managed service, or advisory support for your own team.

Talk to us

If you are deciding where to start with AI, trying to move a pilot into production, or need an outside review of an AI system's data, security or governance, we would be glad to hear from you. Email hello@ps3g.com.

PS3G Inc.

1201 N Market Street, Suite 111, Wilmington, DE 19801

ps3g.com · hello@ps3g.com

This white paper is provided for general information. It is not legal, security or compliance advice for any specific organization. Frameworks and laws referenced (NIST AI RMF 1.0, ISO/IEC 42001, EU AI Act, Colorado SB 24-205, HIPAA, SOC 2, OWASP Top 10 for LLM Applications) should be consulted in their current form. Examples are hypothetical and illustrative. © PS3G Inc.