

AND Digital

The Platform Layer
Building the foundations for governed AI at scale

A Strategic Case Paper by
Gilles Baillet, Head of Technology, Cloud Practice & AI Products
Richard Bovey, Chief for Data

September 2026

A companion paper to AI's Next Frontier, The Trust Imperative, The Build Layer and The Operating Layer

The platform constraint

Consider an organisation in which several teams are building agents over the same internal documents. Each team creates its own retrieval pipeline, applies its own access rules and decides how to check the information returned. Each solution may work in isolation. When a permission changes or a source becomes unreliable, the organisation has several places to investigate and no shared service through which the problem is resolved.

This is an illustration, but the constraint is practical. Adding a more capable model does not resolve fragmented access, inconsistent context or unclear ownership. Those are platform responsibilities, and their consequences grow as agents take on more work.

The platform must give teams a consistent way to deploy and operate agents within agreed boundaries. It must connect each agent's authority to the data it can use, the actions it can take, the cost it incurs and the person who remains accountable. As the volume and speed of agent activity increase, manual oversight cannot keep up. Shared platform services must enforce those boundaries automatically and provide a supported path that teams can reuse. Without them, each team builds its own automation and controls, multiplying the fragmentation described above. These capabilities are essential wherever workloads run, whether in public cloud, private infrastructure or a hybrid estate.

This paper complements the series' work on organisational design, governed AI and software delivery. It explains the shared technical capabilities that allow those arrangements to operate at scale, and how cloud and platform engineering can supply them economically.

Four planes that work together

AND's Enterprise AI Target Architecture describes the platform through four connected planes. User surfaces are the applications and interfaces through which people request work and review results. Intelligence and runtime provide models, orchestration and access to tools. Data and semantics supply the information, definitions and context those systems use. Cloud infrastructure provides the compute, networking, storage and governed environments on which they run. Appendix 1 summarises their responsibilities.

The dependencies run across the whole platform. A user's permissions must survive retrieval and tool use. A runtime decision must be traceable to the information used. A request that requires human approval must return to a person with enough context to decide. Infrastructure alone cannot provide this, although its identity, isolation and observability services make much of it enforceable.

Permission-aware access, checks on context and traceability therefore belong in this paper. The CDO in the Agentic Era explores the underlying data disciplines in more detail. The deployment decision remains separate: public cloud, private infrastructure and hybrid estates can support governed AI when the required capabilities are present.



Platform capabilities for governed AI

Applications already initiate work. Agents extend the range of actions a system can select and take within delegated authority, often across several systems. Five shared capabilities make that authority governable. They require integration with each agent and workflow, even where the platform supplies the common services.

Identity and delegated authority

Each agent needs an identifiable presence, recorded with its purpose, permissions, status and named accountable owner. Access must reflect the work it is authorised to perform: the data it can read, actions it can take and decisions it must escalate. Existing service identities can support this when ownership and permissions are explicit. Shared credentials that obscure which agent acted undermine accountability.

Controls during execution

The platform must check actions against policy while an agent works, it must block and record prohibited activity, and it must allow a running agent to be restricted, isolated or stopped safely. Provisioning controls remain necessary, but an approved environment does not make every subsequent action acceptable. Access, spending limits and approval requirements must remain enforceable during execution. Teams need evidence from testing that these controls work, including when a dependency fails.

Permission-aware data and context

Retrieval must preserve the permissions attached to source data and retain enough provenance to establish what informed an action. The platform should support checks on freshness and quality, with a defined response when information is missing, conflicting or out of date. Shared context services can make these checks reusable. Data owners remain responsible for the data and definitions they provide; a platform cannot guarantee that every source is correct.

Costs linked to successful outcomes

Consumption must be attributable to the agents and workflows that generate it, including retries and unsuccessful executions. This supports cost per successful outcome when combined with infrastructure, human review and recovery costs. Token costs are useful operational inputs. Business value depends on the relationship between total cost, quality, risk and the outcome achieved. A higher cost may be justified by a better result or stronger safeguards.

Recovery and human handover

Failures and requests outside an agent's authority need a defined response. Recovery must avoid repeating actions that have already taken effect. Human handover should preserve the state of the work, the actions taken and the reason for escalation. A named person or responsible team must be able to continue with enough context to decide what happens next. Stopping an agent prevents further action. Any changes it has already made may still need to be reviewed and corrected.

In the opening illustration, these capabilities would allow teams to reuse a governed retrieval service, identify the agent making each request, enforce its access and trace the information returned. A failed permission check would stop that request and create a record. A failed context check would trigger the agreed recovery or human review path. The platform would make a consistent response available across teams, while each workflow owner remained accountable for its use.

The accountability boundary is explicit. Business and agent owners remain accountable for outcomes and behaviour. The platform team is accountable for the controls and services it provides, including their reliability and the evidence that they operate. Making logging or identity a shared service does not transfer ownership of an agent's decisions to that team.



PACE makes the platform dependable

The five capabilities describe what the platform must provide. PACE describes the conditions through which an organisation makes that provision dependable: Platform leverage, Architecture fit, Control in the foundation and Economic honesty. It is AND's practical framework for identifying gaps and deciding where to act.

Platform leverage

A shared platform should remove work that teams would otherwise repeat. An internal developer platform provides a paved road: supported patterns for common tasks such as creating a service, connecting a data source or deploying an agent. Identity, logging, cost attribution and policy enforcement are included in those patterns. Teams can deliver within agreed boundaries without rebuilding the foundations each time.

The platform itself needs product ownership. Its team should understand where users struggle, measure adoption and time to complete common tasks, and improve services through feedback. A catalogue of tools that teams avoid creates little value. Exceptions need a supported route with equivalent controls, so that an unusual requirement does not become a reason to bypass governance.

Gartner forecast that 80% of large software engineering organisations would establish platform engineering teams by 2026, up from 45% in 2022 [1]. That is an adoption forecast, not evidence that every platform investment succeeds. The test is whether the platform helps teams deliver dependable outcomes with less repeated effort.

Architecture fit

Workloads should be matched to the service model and deployment location that suit their needs. Demand variability, latency, data residency, resilience, skills and whole-life cost all matter. Elastic services can suit intermittent demand; stable workloads may justify committed capacity or owned infrastructure. Managed services can remove operational work, while creating dependencies that must be understood.

For agents, this includes where context is held, how tools are reached and how permissions are enforced across system boundaries. A convenient model endpoint does not settle those questions. Architecture fit makes the placement decision explicit and preserves the required controls wherever the workload runs.

Control in the foundation

Landing zones provide governed account and network structures. Infrastructure and policy expressed as code make identity, segmentation, encryption and logging part of the provisioned environment. These defaults reduce the work left to individual teams and make departures easier to detect.

Control must then continue through execution. The platform needs to enforce delegated permissions, approval requirements and consumption limits, with tested isolation, safe stop and recovery. Changes to policy must reach running services through a controlled process. The Trust Imperative supplies the governance principles; the platform provides mechanisms and evidence that make them operational. The forthcoming Security Layer develops the security responsibilities in more detail.

Economic honesty

FinOps brings engineering, finance and the business into decisions about consumption and value. Durable teams own services across their life, including their operating costs. Shared platform costs need a consistent allocation method so that local savings do not conceal costs elsewhere. For an agentic workflow, the business owner must define a successful outcome and the quality and risk conditions attached to it. Total cost includes the work required to produce those outcomes, including failed attempts, retries, review and recovery. Lower unit cost is useful only when the result remains acceptable. PACE asks whether the organisation can explain and justify that relationship, then use it to guide investment.

Cloud choices and the existing estate

Cloud offers many organisations a practical way to assemble these capabilities. Elastic capacity accommodates variable demand; managed services can reduce operational work; and environments that can be provisioned quickly support delivery. Access to inference and accelerator services adds to that case, subject to capacity, location and cost constraints. These benefits depend on how workloads and teams use the services available.

Cloud economics also depends on workload fit. In October 2024, 37signals reported almost $2 million in annual cloud-bill reductions after moving seven applications from AWS to ts own hardware in 2023. Its remaining cloud spend included S3 storage; the larger five-year savings figure was a projection [2]. Dropbox's 2018 filing reported infrastructure cost reductions of $39.5 million in 2016 and a further $35.1 million in 2017 following its infrastructure optimisation programme [3].

Capital One chose a different route, completing its exit from eight data centres in 2020. AWS's customer account describes rebuilding 80% of nearly 2,000 applications and reducing the average development environment build time from three months to minutes [4]. These cases illustrate different placement decisions and reported benefits. They do not establish the effectiveness of an agentic platform or make either deployment model universally preferable.

Switching costs also need precision. AWS's exit programme, announced in March 2024, requires a request and approval before migration credits are provided [5]. Separately, the EU Data Act removes switching charges, including data-egress charges, from 12 January 2027 for services within its scope [6]. This does not eliminate application changes, operational transition work or all contractual costs. Dependencies still belong in the architectural decision.

Most organisations must make these choices within a brownfield estate. Moving a workload unchanged can meet a data-centre exit deadline, but it does not automatically capture elasticity or managed-service benefits. Keeping permanently allocated capacity at metered rates may increase cost without changing how the organisation works. Rehosting can be a deliberate first step when the subsequent work is funded and owned.

Modernisation should follow business value. Refactor where the expected benefit justifies the effort; replatform where a managed service removes a material burden; retain systems that still serve their purpose; and retire those that no longer do. Replacing a system may be more useful than rebuilding it. The portfolio needs explicit decisions, rather than one migration strategy applied to every workload.

Moving workloads out of public cloud does not prevent an organisation from scaling AI. It changes which capabilities the organisation must provide and operate itself. A private or hybrid estate still needs governed access, runtime controls, recovery and outcome economics. The relevant question is whether the chosen arrangement can sustain those capabilities at the required quality and cost.

A practical route to progress

Start with a business workflow whose outcome, owner and boundaries can be stated. Assess the platform services it depends on, then close the gaps that prevent safe operation. Establish the shared identity, context, runtime controls and cost attribution needed for that workflow before extending them to others. Teams can develop capabilities together; they do not need to pass through a fixed sequence of cloud adoption stages.

The first implementation should make the next one easier. Measure whether teams reuse the shared services, whether exceptions are handled consistently and whether operating evidence supports wider deployment. Expand on the basis of those results. Building every possible platform service in advance risks creating a costly product with few users; allowing every team to improvise recreates the fragmentation described at the start.

Infrastructure choices alone do not establish enterprise AI maturity. The transition from building capabilities to scaling AI, discussed in AI's Next Frontier, also requires redesigned workflows, accountable ownership and organisational adoption. The platform supports that transition by making governed execution repeatable. It cannot complete the organisational work on its own.

A diagnostic for your organisation

Use PACE to examine a live workflow or one approaching production. Ask for evidence from the people responsible, rather than a general statement that the estate is ready.

PACE condition	Evidence to ask for
Platform leverage	Which shared services does this workflow use? Can another team adopt them through a supported path, and where does repeated manual work remain?
Architecture fit	Why do the workload and its data run here? Can we justify the choice against residency, latency, resilience, skills and whole-life cost?
Control in the foundation	Who owns the agent, what can it do, and can we demonstrate permission enforcement, a safe stop and a handover with state preserved?
Economic honesty	Which shared services does this workflow use? Can another team adopt them through a supported path, and where does repeated manual work remain?

An uncertain answer identifies work to do. Agree its owner, the evidence that will show it is complete and the conditions under which the workflow can expand. This gives leaders a practical basis for deciding where to invest and what needs to change.

The AND way

At AND Digital, Guide, Build, Equip provides a practical route from that assessment to a platform the client can operate and improve.

Guide establishes the business outcome, current constraints and target architecture. It uses PACE to prioritise gaps and make workload-placement, ownership and investment decisions explicit. The result is a sequence of work tied to the client's intended outcomes.

Build implements and integrates the shared capabilities, including the paved road, governed context access, runtime controls and cost attribution. Migration and modernisation are sequenced around the workloads that justify them, with recovery and handover tested before the scope of autonomous action grows
.
Equip develops the client's ability to own and improve the platform. That includes engineering skills, operational responsibilities and a working relationship between platform, data and business teams. The platform should remain useful as the organisation's needs change.

AND Harness is AND's implementation of the governed delivery and runtime pattern. Suitable client capabilities can meet the same requirements, and AND Flow operates with or without Harness, as The Build Layer sets out. The engagement starts with the capabilities and outcomes required, with the implementation chosen to fit.

The immediate decision is which workflow the organisation is ready to support, what prevents it doing so dependably and which shared capabilities will remove that constraint. PACE gives leaders a way to make that decision with evidence. Cloud and platform engineering give them practical means to act on it, so that the next agent inherits a foundation the organisation can trust and afford to operate.

References

1. Gartner. Platform Engineering. Research hub, accessed 21 September 2026. Forecast of platform engineering team adoption by 2026. https://www.gartner.com/en/infrastructure-and-it-operations-leaders/topics/platform-engineering

2. David Heinemeier Hansson. Our cloud-exit savings will now top ten million over five years. October 2024. Company-reported annual bill reduction and separate five-year projection. https://world.hey.com/dhh/our-cloud-exit-savings-will-now-top-ten-million-over-five-years-c7d9b5bd

3. Dropbox, Inc. Form S-1, filed 23 February 2018, pp. 71 and 73. Year-on-year infrastructure cost reductions in 2017 and 2016. https://www.sec.gov/Archives/edgar/data/1467623/000119312518055809/d451946ds1.htm

4. AWS. Capital One Completes Migration from Data Centers to AWS. Customer case study describing its 2020 data-centre exit and reported delivery benefits. https://aws.amazon.com/solutions/case-studies/capital-one-all-in-on-aws/

5. Sébastien Stormacq. Free data transfer out to internet when moving out of AWS. AWS News Blog, 5 March 2024, updated 30 September 2025. https://aws.amazon.com/blogs/aws/free-data-transfer-out-to-internet-when-moving-out-of-aws/

6. European Commission. Data Act explained. Section on switching between data processing services. Accessed 21 September 2026. https://digital-strategy.ec.europa.eu/en/factpages/data-act-explained

Companion papers
AND Digital. AI's Next Frontier, updated June 2026; The Trust Imperative, May 2026; The Build Layer, June 2026; The Operating Layer, July 2026. Related companion on data: The CDO in the Agentic Era. Forthcoming companion on security: The Security Layer.

Appendix 1

The four planes of the Platform Layer

A simplified view of AND's Enterprise AI Target Architecture. The Operating Layer defines responsibilities and ways of working. The delivery capabilities described in The Build Layer build and change the platform and the applications it supports.

Platform plane	Shared capability	Connection to the other planes
User surfaces	Applications, assistants and interfaces for requesting work, approvals and human handover	Carry the user's identity and intent into the runtime; return results, evidence and requests for intervention
Intelligence and runtime	Model access, orchestration, context retrieval, tool use and execution controls	Use authorised data and tools; enforce limits; record actions, consumption and recovery state
Data and semantics	Data products, shared definitions, catalogues, provenance and governed storage	Supply permission-aware context and traceable meaning; expose quality and freshness information to the runtime
Cloud infrastructure	Compute, networking, storage, governed environments, automation and self-service	Provide the hosting, identity, isolation and observability services on which the other planes depend

Identity, policy enforcement, audit, observability, approval and cost controls span the four planes. Their implementation is shared; accountability remains with the owners of the services, agents and business outcomes concerned.
Cloud infrastructure is the plane name used in the target architecture. It does not prescribe public-cloud hosting. The same responsibilities must be met in private or hybrid deployments, and technology choices depend on the workload and the client's requirements.