

When the Model Moves: Entrepreneurial Adaptation to Foundation-Model Dependency in AI Startups

Fahad Aljadaan

Independent Researcher, Kuwait City, Kuwait

F.K.Aljadaan@gmail.com

Abstract: Foundation models allow artificial intelligence (AI) startups to develop sophisticated products without owning the underlying model, but this speed creates dependence on providers that control model behaviour, availability, pricing, safety settings and lifecycle decisions. This paper examines how the AI venture ecosystem publicly constructs that dependence as manageable and which adaptation strategies it makes available to downstream ventures. The study uses qualitative content analysis of 15 organisation-controlled online sources published by model providers, AI gateways and routers, developer frameworks, and observability and evaluation platforms. Sources were selected using explicit criteria concerning ecosystem role, public accessibility, recency and substantive treatment of model lifecycle, switching, routing, fallback, evaluation or integration. Provider materials are treated as strategic discourse rather than neutral evidence of implementation; interpretations are therefore bounded, compared across ecosystem roles and contextualised using recent independent research on platform power, transparency, competition and AI business models. The findings identify model volatility as the operating condition and five responses: multi-model hedging, abstraction-layer building, failure routing and fallback, evaluation-based capability monitoring, and workflow ownership. Together these responses form a dependency-management architecture. The paper defines workflow ownership as venture-specific control over the customer process, data context, integration logic, evaluation criteria, governance and assurance surrounding model inference, so that customer value can persist when the upstream model changes. The analysis develops the concept of resilient dependence: ventures may remain reliant on external models while reducing the ability of any single provider or model version to determine continuity. Economically, the architecture trades additional integration, evaluation and governance costs for lower exposure to supplier power, service interruption, margin volatility and forced migration. The study contributes a process model for research on AI entrepreneurship and a practical framework for assessing whether a downstream venture can continue delivering and capturing value when its upstream model moves.

Keywords: AI Entrepreneurship; Foundation Models; Platform Dependence; Dynamic Capabilities; Resilient Dependence; Workflow Ownership

1. Introduction

Foundation models have lowered the cost and time required to build artificial intelligence (AI) products. Startups can access large language models, multimodal systems and hosted inference through application programming interfaces rather than developing the underlying technology. This expands entrepreneurial opportunity, but it also places a critical part of the product outside the venture's control. A provider can retire a model, alter prices or rate limits, change safety behaviour, or release a successor whose outputs differ from the version on which the product was designed.

Provider documentation shows that such movement is routine rather than exceptional. OpenAI maintains deprecation notices, Anthropic publishes migration guidance between Claude generations, and Google distinguishes stable, preview, latest and deprecated Gemini models (OpenAI, 2026; Anthropic, 2026; Google AI for Developers, 2026). For a downstream venture, an apparently technical change can affect customer promises, unit economics, latency, reliability and legitimacy. Prompt logic may need to be revised, evaluations rerun, fallbacks introduced and contracts reconsidered.

Recent research reinforces the strategic character of this problem. Generative AI is increasingly analysed as a multi-level system in which application value depends on models, infrastructure and organisational integration rather than on a model alone (Feuerriegel et al., 2024). Innovation-management and business-model research likewise emphasises ecosystem governance, value creation and organisational boundaries (Mariani and Dwivedi, 2024; Jorzik et al., 2024). Most directly, Nzembayie and Urbano (2026) show how generative-AI platforms enable entrepreneurship while creating dependence on non-substitutable, platform-governed capabilities. Competition research also warns that the foundation-model layer may concentrate power and shape downstream innovation (Schrepel and Pentland, 2025).

Digital entrepreneurship research explains how digital technologies reshape entrepreneurial uncertainty and action (Nambisan, 2017). Platform research further shows that complementors gain access to shared infrastructure while remaining exposed to interfaces and governance controlled by platform owners (Gawer and Cusumano, 2014; Jacobides, Cennamo and Gawer, 2018). Cutolo and Kenney (2021) describe platform-

dependent entrepreneurs as actors facing structural power asymmetries. Foundation-model markets intensify this condition: the upstream actor does not only govern access or distribution; it supplies part of the product's cognitive performance.

This paper defines foundation-model dependency as a condition in which a venture's product performance, cost structure, reliability and strategic direction depend materially on external model providers that the venture does not control. It asks: How does the AI venture ecosystem publicly construct foundation-model dependency as a manageable condition, and which adaptation strategies does it institutionalise? The paper develops an ecosystem-level typology and interprets it through resource dependence and dynamic capabilities. Its central argument is that AI ventures build resilient dependence by embedding sensing, switching and reconfiguration into technical architecture while locating defensibility in the workflow surrounding the model.

2. Theoretical Background

2.1 Resource and Platform Dependence

Resource dependence theory holds that organisations become vulnerable when another actor controls a resource that is important and difficult to substitute (Pfeffer and Salancik, 1978). Young ventures are particularly exposed because they have limited slack, bargaining power and established routines. A foundation model is an unusual critical resource: it can simultaneously determine output quality, reasoning capability, response style, latency, safety behaviour and serving cost. The provider is therefore both an enabler of venture creation and a continuing source of strategic constraint.

Platform ecosystems create similar tensions. Shared standards and interfaces can accelerate innovation, but complementors remain exposed to rule changes, interface restrictions, appropriation and shifting participation economics (Gawer and Cusumano, 2014; Jacobides, Cennamo and Gawer, 2018). Foundation-model dependency is a deeper form of this relationship because changes can enter the functional behaviour of the downstream product. The relevant shock is not only a commission or ranking change; it can be a new refusal pattern, altered reasoning performance, different token use or model retirement.

Recent independent evidence strengthens the need for caution toward provider-centred accounts. The Foundation Model Transparency Index documents continuing opacity around downstream impact and other areas despite improved disclosure (Bommasani et al., 2024). Research on competition highlights the possibility that a small number of foundation-model firms can shape access, prices and innovation conditions for downstream actors (Schrepel and Pentland, 2025), while large-scale analysis of open-model activity shows that openness can redistribute participation without eliminating concentration or intermediary power (Longpre et al., 2025). Dependency management therefore concerns not only technical substitutability but also bargaining power, information asymmetry and value capture.

2.2 Dynamic Capabilities Under Model Volatility

Dynamic capabilities describe how firms sense changes, seize opportunities and reconfigure resources in changing environments (Teece, Pisano and Shuen, 1997; Teece, 2007). In high-velocity settings, these capabilities often take the form of identifiable processes rather than one universal routine (Eisenhardt and Martin, 2000). This lens is useful for AI startups because model markets change frequently and adaptation can be encoded in software architecture.

Evaluation and observability can support sensing by exposing regressions, refusal changes, latency and cost. Routing, fallback and access to alternative models can support seizing by creating immediate options when conditions change. Common interfaces and workflow-specific assets can support reconfiguration by allowing the venture to replace or rearrange the model layer while preserving its own application logic. The empirical question is therefore not simply whether a venture is dependent, but which publicly available architectures make that dependence more resilient.

2.3 Business-model and Economic Consequences

AI-driven business-model innovation involves more than adding a new technical feature; it changes value propositions, activities, partners, cost structures and value-capture mechanisms (Kanbach et al., 2024; Jorzik et al., 2024). For an application-layer startup, foundation-model dependence creates variable inference costs, migration expenditure, duplicated testing, possible downtime and exposure to unilateral changes in upstream

terms. At the same time, building gateways, evaluations and assurance routines requires investment before their value is visible. Resilience is therefore an economic allocation problem: the venture incurs option-building costs today to reduce the expected cost and bargaining impact of future provider movement.

3. Method

3.1 Design, Unit of Analysis and Source Rationale

The study uses qualitative content analysis of public online materials. The unit of analysis is the organisation-controlled source artifact: documentation, migration guidance or product material describing model lifecycle, switching, routing, fallback, observability, evaluation or integration. These materials are appropriate because the research question concerns how the ecosystem represents dependency and adaptation to prospective users. They are not treated as neutral evidence that every advertised feature is effective, widely adopted or economically beneficial.

Provider discourse is analytically relevant because it helps define the repertoire of actions made visible and legitimate to downstream ventures. At the same time, the organisations have commercial incentives to portray dependence as manageable. The analysis therefore separates the stated mechanism from the paper's interpretation of its entrepreneurial function. A higher-order response was retained only when supported by a concrete operational description and repeated across more than one organisation or ecosystem role. Recent independent research on transparency, platform power, competition and business models was used for contextual triangulation and to identify counter-arguments; it was not counted as part of the 15-source empirical corpus.

3.2 Sample Selection

Fifteen organisations were selected purposively in July 2026 using maximum variation across the infrastructure that connects foundation models to downstream applications. A source was included only when it met all five criteria: (1) it was official and organisation-controlled; (2) it was publicly accessible in English; (3) it was current or actively maintained at the time of retrieval; (4) it contained substantive, mechanism-level material on model lifecycle, switching, routing, fallback, evaluation, observability or application integration; and (5) the organisation directly supplied or governed a technical component used by downstream AI applications.

General commentary, end-user application marketing, duplicate pages and sources containing only broad claims such as 'more reliable' or 'provider agnostic' without operational explanation were excluded. Sampling sought role diversity rather than statistical representation: model providers reveal the upstream condition; gateways and routers expose switching and continuity mechanisms; development frameworks show application-layer separation; and observability or evaluation platforms reveal how change is detected. The final set of 15 was reached when additional sources repeated the existing higher-order categories rather than adding a new response.

Table 1:

Ecosystem Role	Organisations and Material	Evidence Coded
Model providers	OpenAI; Anthropic; Google AI for Developers	Deprecation, migration guidance and model lifecycle categories
Gateways, routers and unified access	Vercel; LiteLLM; Portkey; OpenRouter; Helicone; Eden AI; Requesty	Multi-provider access, routing, fallback, retries, budgets, health checks, cost and latency
Development frameworks	LangChain; LlamaIndex	Common interfaces, provider integrations and application-layer separation
Observability and evaluation	Langfuse; Humanloop; Braintrust	Tracing, prompt comparison, evaluations, regression detection and production monitoring

3.3 Coding and Analytic Procedure

Coding proceeded in three stages. First-cycle descriptive codes captured deprecation, retirement, migration, provider switching, common interfaces, retries, fallback, routing, load balancing, cost control, latency, prompt comparison, evaluation, regression detection, provider agnosticism, integration and workflow context. Second, related codes were grouped into an operating condition and five higher-order responses. Third, the themes were compared with resource dependence and the sensing-seizing-reconfiguring framework to identify their entrepreneurial function.

The analysis was conducted by a single coder and maintained an audit trail of source URLs, access dates, descriptive codes and theme assignments. Negative implications were recorded alongside claimed benefits, including non-equivalence between models, intermediary dependence, additional testing costs and the possibility that advertised controls do not reflect production performance. The claims are consequently interpretive and ecosystem-level: they concern a publicly available repertoire of adaptation, not causal effects on venture performance.

4. Findings

The analysis identifies model volatility as the operating condition and five responses. Table 2 presents the relationship between public mechanisms, entrepreneurial functions and dynamic-capability roles.

Table 2:

Condition or Response	Public Mechanism	Entrepreneurial Function	Capability Role
Model volatility	Deprecation, migration and changing lifecycle status	Creates dependency pressure and exposes unit economics	Environmental turbulence
Multi-model hedging	Access to several models or providers	Diversifies supplier dependence and creates options	Seizing
Abstraction layers	Common interfaces and provider integrations	Protects application logic and lowers direct switching costs	Reconfiguring
Failure routing	Retries, routing rules and fallback chains	Buffers outages, constraints and revenue interruption	Seizing
Evaluation and observability	Tracing, evaluations and regression monitoring	Makes drift, cost and quality changes visible	Sensing
Workflow ownership	Data context, integrations, assurance and customer process	Relocates defensibility and value capture around the model	Reconfiguring and renewal

4.1 Model Volatility is Normalised

Model-provider documentation presents retirement and migration as part of normal production use. OpenAI records deprecated models and replacement paths; Anthropic explains changes in model names, parameters, prompting, cost and latency; and Google's lifecycle categories warn that preview or latest versions may change more quickly (OpenAI, 2026; Anthropic, 2026; Google AI for Developers, 2026). These materials frame continuity as conditional rather than guaranteed.

For downstream ventures, improvement can therefore be destabilising. A more capable model may change output structure, refusals, token use or response speed. The entrepreneurial risk is not only market demand but the ability to maintain a promise when upstream cognitive infrastructure moves. Economically, movement can alter gross margins, trigger unplanned engineering and validation expenditure, delay sales or require price and service-level renegotiation. This normalised volatility creates the need for the remaining responses.

4.2 Multi-model Hedging Creates Optionality

Gateways and unified APIs repeatedly present access to multiple models and providers as a core benefit. Vercel, Helicone, Eden AI and Requesty emphasise common access, provider choice, cost controls and model-selection

policies, while OpenRouter and LiteLLM expose routing across alternative endpoints (Vercel, 2026; Helicone, 2026; Eden AI, 2026; Requesty, 2026; OpenRouter, 2026; LiteLLM, 2026). The strategic logic is supplier diversification: dependence remains, but it is spread across external sources.

This converts model multiplicity into operational optionality. A venture can redirect a workload when a provider is unavailable, slow, expensive or unsuitable for a task. The economic value is a reduction in the expected loss from outage or unilateral price change and an improved negotiating position when credible alternatives exist. Yet optionality is not free: connectors, duplicated evaluations, data-governance reviews and minimum spend across providers can raise fixed and coordination costs. Multi-model access is therefore valuable when the avoided disruption exceeds the cost of maintaining tested alternatives.

4.3 Abstraction Layers Protect Venture-Specific Logic

LangChain, LlamaIndex and Vercel's AI SDK describe common interfaces and broad integrations across model providers (LangChain, 2026; LlamaIndex, 2026; Vercel, 2026). Such abstraction separates application logic from provider-specific calls. Without it, migration may require extensive rewriting; with it, more of the venture's orchestration, user interface and domain logic can remain stable when the model changes.

Abstraction is therefore more than engineering convenience. It lowers direct switching costs and supports reconfiguration, but it can also shift dependence to the framework or gateway that supplies the common interface. Lowest-common-denominator design may prevent use of distinctive provider features, while intermediary fees can affect margins. The strategy is strongest when the venture retains portability of its prompts, evaluations, logs and business logic rather than merely replacing one opaque dependency with another.

4.4 Failure Routing Converts Disruption Into Managed Continuity

Portkey, LiteLLM, OpenRouter, Helicone and Requesty describe retries, health checks, routing rules and fallback chains that move requests away from a failed or constrained model (Portkey, 2026; LiteLLM, 2026; OpenRouter, 2026; Helicone, 2026; Requesty, 2026). These mechanisms buffer upstream outages, rate limits and performance problems before they become customer-facing failure.

Fallback is a seizing capability because it enables action under constraint, but it requires prior choices about acceptable substitutes. A lower-cost or faster model may not preserve quality, compliance or output format. Effective routing therefore depends on workflow-level service criteria and evaluation, not merely on the existence of a second endpoint. Its economic role is continuity protection: routing can preserve transactions, user trust and contractual service levels, but poor fallback can create hidden quality costs that emerge later through complaints, rework or liability.

4.5 Evaluation and Observability Provide Sensing Infrastructure

Langfuse, Humanloop and Braintrust present tracing, prompt comparison, evaluations, production monitoring and regression detection as continuous elements of AI application development (Langfuse, 2026; Humanloop, 2026; Braintrust, 2026). Gateways also expose cost, latency and error information. Together, these systems make upstream change visible to the downstream venture.

This is the clearest sensing response. Changing an endpoint does not show whether the replacement still performs the task acceptably. Evaluation routines allow a venture to compare model versions, identify weakened workflows and decide whether prompts, routing rules, prices or customer promises must change. Economically, evaluation converts an uncertain migration into a measurable decision and reduces the risk of silent degradation. It also creates a reusable organisational asset: test cases, acceptance thresholds and production traces accumulate knowledge that remains valuable across providers.

4.6 Workflow Ownership Relocates Defensibility

Workflow ownership is defined here as venture-specific control over the customer process surrounding model inference: problem framing, domain data and context, integration logic, evaluation criteria, governance, human escalation, assurance and the user relationship. It does not mean legal ownership of the customer's process or independence from all external infrastructure. Rather, it means that the venture controls the configuration through which generic model capability becomes a reliable, accountable and economically valuable outcome.

If foundation-model access is externally controlled and widely available, access alone is a weak basis for defensibility. Hedging, abstraction, routing and evaluation allow a venture to remain valuable while changing the underlying model. The venture becomes less reducible to one provider when customers depend on its data connections, operational controls, verified outcomes and embedded process knowledge. Workflow ownership can support recurring revenue, retention and differentiated assurance, but defensibility is durable only when it reflects genuine customer value rather than artificial lock-in.

4.7 The Economic Logic of Resilient Dependence

Across the five responses, resilient dependence is not costless independence. It is a portfolio of real options. The venture pays for integration, monitoring, evaluation, duplicate testing and governance in exchange for lower expected losses from forced migration, downtime, quality drift and supplier price changes. The optimal level is therefore contingent. A low-risk consumer feature may tolerate single-provider dependence; a regulated or mission-critical workflow may justify extensive redundancy and assurance.

The architecture also influences where value is captured. Upstream model providers retain scale economies and control over core capability, while gateways and frameworks may capture value by reducing transaction and switching costs. Downstream startups can defend margins when they convert generic inference into workflow-specific outcomes that customers are willing to pay for and that survive provider substitution. This creates a central tension: the same abstraction that improves portability may commoditise the model layer, increasing pressure on the venture to differentiate through data, integration, distribution, trust and verified performance.

5. Discussion

5.1 Theoretical Contributions

First, the findings extend platform-dependent entrepreneurship. Foundation-model dependency is not only dependence on access, distribution or governance; it is dependence on externally controlled product behaviour. This helps explain why AI ventures may remain vulnerable even when they retain their customers and application interface. It also complements recent work showing that generative-AI platforms simultaneously enable resource access and reproduce power asymmetries (Nzembayie and Urbano, 2026).

Second, the paper extends resource dependence into AI entrepreneurship by treating model capability as external cognitive infrastructure. The response is not full independence but resilient dependence: diversification, visibility and switching reduce the extent to which one provider can determine venture continuity. The concept preserves the relational insight of resource dependence while recognising that substitution is partial, costly and often mediated by other platforms.

Third, the findings make dynamic capabilities architectural. Sensing appears in evaluations and observability; seizing appears in routing, fallback and multi-model choice; reconfiguration appears in abstraction and the relocation of value to workflow-specific assets. The technical stack is therefore part of the organisation's adaptation capability, not merely an implementation detail. Repeated evaluations, routing rules and integration patterns can become routines through which a young venture learns under technological volatility.

Fourth, the paper links dependency management to business-model economics. Resilience mechanisms affect costs, supplier bargaining power, revenue continuity and the location of defensibility. This addresses a gap in technology-centred accounts by showing that architecture is also a value-capture decision. The venture must decide which dependencies to diversify, which to accept and which complementary assets to own.

5.2 A Process Model of Resilient Dependence

The themes form a process model. Anticipated or realised model movement creates migration pressure. Evaluation and observability reveal the consequences; multi-model access and routing create response options; abstraction reduces the cost of changing the stack; and workflow ownership preserves the venture's distinctive value. The outcome is not freedom from foundation models but a more resilient relationship with them.

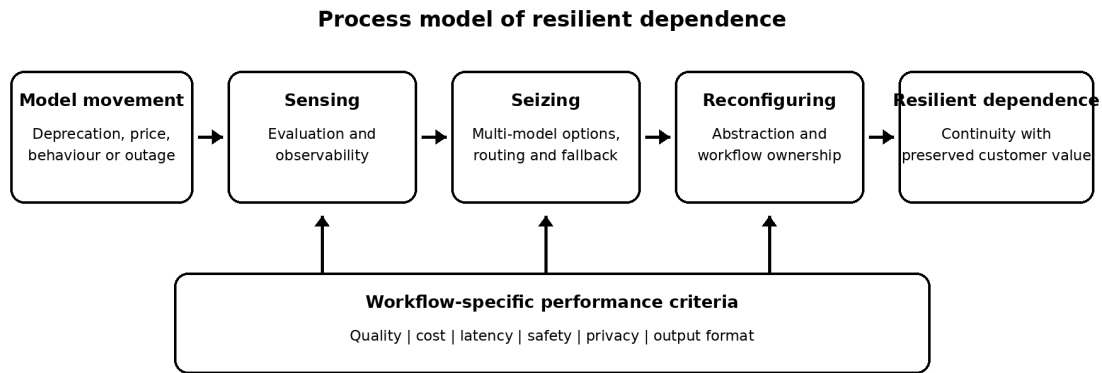


Figure 1: Process model of resilient dependence

Figure 1 clarifies the sequence and its boundary condition. Architectural flexibility is valuable only when the venture knows what must remain invariant. Quality, latency, cost, safety, privacy and output format may matter differently across workflows. These criteria guide sensing, determine acceptable fallback options and define what reconfiguration must preserve. The process is iterative: once the model or workflow changes, production evidence feeds back into evaluation and may trigger another cycle.

6. Implications

For founders, model choice should be treated as a strategic dependency decision rather than a one-time engineering choice. Due diligence should ask whether critical workflows can switch models, whether fallback options are tested, whether regressions and costs are monitored, and what the venture owns beyond prompts and API access. The relevant metric is not simply the number of supported models but the time, cost and quality loss associated with a realistic migration.

For investors and accelerators, dependency architecture can be a signal of venture maturity. A useful assessment is whether customer value and gross margin would survive a provider change. Evidence may include portable evaluation sets, tested fallback routes, concentration of inference spend, contractual exposure, customer-specific integrations and the share of value located in workflow assets rather than raw model access.

For policymakers and ecosystem builders, support should extend beyond access to models and include evaluation sandboxes, interoperability, open and portable infrastructure, procurement guidance, data governance and assurance capabilities. Competition policy should also consider how upstream concentration and intermediary control affect downstream experimentation and value capture (Schrepel and Pentland, 2025). Such support can reduce thin, captive positions without assuming that open models or multi-provider access automatically remove dependence.

7. Limitations and Future Research

The study analyses public organisation-controlled materials rather than internal decisions or verified technical performance. Marketing and documentation may selectively emphasise controllability, understate implementation difficulty and omit failures. The source design is legitimate for analysing the public repertoire through which dependency is framed, but it cannot establish adoption, effectiveness or causal impact. Contextual triangulation with independent research reduces the risk of accepting provider claims at face value, but it does not replace independent empirical observation.

The purposive sample is infrastructure-oriented and cannot represent all AI startups, while single-coder analysis limits reliability claims. The sample also captures a rapidly changing ecosystem at one point in time, and several organisations are themselves intermediaries with incentives to expand the perceived need for their services. The typology and process model are therefore exploratory.

Future research should interview founders who have experienced migrations, outages, safety changes and pricing shocks; compare advertised mechanisms with logs and post-migration performance; and contrast single-provider, multi-provider and open-weight architectures. Longitudinal studies could test whether dependency-management capability predicts survival, enterprise adoption, growth or gross-margin stability. Further work should examine when abstraction creates new intermediary dependence, when workflow ownership becomes

sufficiently proprietary to create durable advantage, and how the optimal resilience investment varies by regulation, workflow criticality and venture stage.

8. Conclusion

AI startups build with powerful models that they do not own and whose behaviour can change. This paper identifies model volatility as the operating condition and five ecosystem responses: multi-model hedging, abstraction layers, failure routing, evaluation and observability, and workflow ownership. Together, these mechanisms constitute a dependency-management architecture.

The central implication is that resilience comes from controlling the relationship between model capability and customer value. The most defensible downstream ventures may not own the foundation model, but they can own the workflow, data context, evaluation routines, integrations and assurance layer that allow the product to keep working when the model moves. Resilient dependence is therefore both a dynamic capability and an economic design choice: it balances the cost of maintaining options against the risk that an upstream provider can interrupt continuity, alter margins or appropriate a larger share of value.

Acknowledgements

None.

Ethics Declaration

Ethical clearance was not required because the study analysed publicly available, organisation-controlled online materials and did not involve human participants, private data or interaction with individuals.

AI Declaration

Generative AI tools were used for language editing, structural refinement, and formatting assistance under the author's direction. All study design, source selection, coding decisions, interpretation, verification, and final responsibility remain with the author.

References

- Anthropic (2026) 'Migration guide: Migrating to the latest Claude models', Claude Platform Docs, <https://platform.claude.com/docs/en/about-claude/models/migration-guide> (accessed 8 July 2026).
- Bommasani, R., Klyman, K., Kapoor, S., Longpre, S., Xiong, B., Maslej, N. and Liang, P. (2024) 'The Foundation Model Transparency Index v1.1: May 2024', arXiv:2407.12929.
- Braintrust (2026) 'Ship quality AI at scale', <https://www.braintrust.dev/> (accessed 8 July 2026).
- Cutolo, D. and Kenney, M. (2021) 'Platform-dependent entrepreneurs: Power asymmetries, risks, and strategies in the platform economy', *Academy of Management Perspectives*, Vol. 35, No. 4, pp. 584-605. doi:10.5465/amp.2019.0103.
- Eden AI (2026) 'One API to route best AI models', <https://www.edenai.co/> (accessed 8 July 2026).
- Eisenhardt, K.M. and Martin, J.A. (2000) 'Dynamic capabilities: What are they?', *Strategic Management Journal*, Vol. 21, Nos. 10-11, pp. 1105-1121. doi:10.1002/1097-0266(200010/11)21:10/11<1105::AID-SMJ133>3.0.CO;2-E.
- Feuerriegel, S., Hartmann, J., Janiesch, C. and Zschech, P. (2024) 'Generative AI', *Business & Information Systems Engineering*, Vol. 66, pp. 111-126. doi:10.1007/s12599-023-00834-7.
- Gawer, A. and Cusumano, M.A. (2014) 'Industry platforms and ecosystem innovation', *Journal of Product Innovation Management*, Vol. 31, No. 3, pp. 417-433. doi:10.1111/jpim.12105.
- Google AI for Developers (2026) 'Gemini API models', <https://ai.google.dev/gemini-api/docs/models> (accessed 8 July 2026).
- Helicone (2026) 'AI Gateway overview', <https://docs.helicone.ai/gateway/overview> (accessed 8 July 2026).
- Humanloop (2026) 'Compare and debug prompts', <https://humanloop.com/docs/guides/evals/comparing-prompts> (accessed 8 July 2026).
- Jacobides, M.G., Cennamo, C. and Gawer, A. (2018) 'Towards a theory of ecosystems', *Strategic Management Journal*, Vol. 39, No. 8, pp. 2255-2276. doi:10.1002/smj.2904.
- Jorzik, P., Klein, S.P., Kanbach, D.K. and Kraus, S. (2024) 'AI-driven business model innovation: A systematic review and research agenda', *Journal of Business Research*, Vol. 182, 114764. doi:10.1016/j.jbusres.2024.114764.
- Kanbach, D.K., Heiduk, L., Blueher, G., Schreiter, M. and Lahmann, A. (2024) 'The GenAI is out of the bottle: Generative artificial intelligence from a business model innovation perspective', *Review of Managerial Science*, Vol. 18, No. 4, pp. 1189-1220. doi:10.1007/s11846-023-00696-z.
- LangChain (2026) 'Providers and models', <https://docs.langchain.com/oss/python/concepts/providers-and-models> (accessed 8 July 2026).
- Langfuse (2026) 'Open-source AI engineering platform', <https://langfuse.com/> (accessed 8 July 2026).

- LiteLLM (2026) 'Routing and load balancing', <https://docs.litellm.ai/docs/routing-load-balancing> (accessed 8 July 2026).
- LlamaIndex (2026) 'Integration packages and documentation', https://github.com/run-llama/llama_index (accessed 8 July 2026).
- Longpre, S., Akiki, C., Lund, C., Kulkarni, A., Chen, E., Solaiman, I., Ghosh, A., Jernite, Y. and Kaffee, L.-A. (2025) 'Economies of open intelligence: Tracing power and participation in the model ecosystem', arXiv:2512.03073.
- Mariani, M. and Dwivedi, Y.K. (2024) 'Generative artificial intelligence in innovation management: A preview of future research developments', *Journal of Business Research*, Vol. 175, 114542. doi:10.1016/j.jbusres.2024.114542.
- Nambisan, S. (2017) 'Digital entrepreneurship: Toward a digital technology perspective of entrepreneurship', *Entrepreneurship Theory and Practice*, Vol. 41, No. 6, pp. 1029-1055. doi:10.1111/etap.12254.
- Nzembayie, K.F. and Urbano, D. (2026) 'Generative AI platforms as institutional catalysts of digital entrepreneurship: Enablement, dependence and power dynamics', *Technology in Society*, Vol. 84, 103074. doi:10.1016/j.techsoc.2025.103074.
- OpenAI (2026) 'Deprecations', OpenAI API Documentation, <https://developers.openai.com/api/docs/deprecations> (accessed 8 July 2026).
- OpenRouter (2026) 'Provider routing and model fallbacks', <https://openrouter.ai/docs/guides/routing/provider-selection> (accessed 8 July 2026).
- Pfeffer, J. and Salancik, G.R. (1978) *The External Control of Organizations: A Resource Dependence Perspective*, Harper and Row, New York.
- Portkey (2026) 'Fallbacks', Portkey Docs, <https://portkey.ai/docs/product/ai-gateway/fallbacks> (accessed 8 July 2026).
- Requesty (2026) 'AI Gateway and LLM router', <https://www.requesty.ai/> (accessed 8 July 2026).
- Schrepel, T. and Pentland, A.S. (2025) 'Competition between AI foundation models: Dynamics and policy recommendations', *Industrial and Corporate Change*, Vol. 34, No. 5, pp. 1085-1103. doi:10.1093/icc/dtae042.
- Teece, D.J. (2007) 'Explicating dynamic capabilities: The nature and microfoundations of (sustainable) enterprise performance', *Strategic Management Journal*, Vol. 28, No. 13, pp. 1319-1350. doi:10.1002/smj.640.
- Teece, D.J., Pisano, G. and Shuen, A. (1997) 'Dynamic capabilities and strategic management', *Strategic Management Journal*, Vol. 18, No. 7, pp. 509-533. doi:10.1002/(SICI)1097-0266(199708)18:7<509::AID-SMJ882>3.0.CO;2-Z.
- Vercel (2026) 'AI SDK and AI Gateway documentation', <https://vercel.com/docs/ai-sdk> and <https://vercel.com/docs/ai-gateway> (accessed 8 July 2026).