GenAI Build vs. Buy Decision Matrix

A practical framework for deciding whether to build, buy, or partner for GenAI capabilities. Based on evaluating 30+ build vs. buy decisions across enterprise AI product development. The framework covers: evaluation criteria, scoring rubric, decision tree, 3-year cost modeling, and vendor evaluation process.
The most common mistake: teams default to 'buy' for speed and then discover 18 months later that the vendor can't meet their regulatory, customization, or integration requirements. The second most common mistake: teams default to 'build' for control and underestimate maintenance cost and talent requirements. This framework makes the tradeoffs explicit before you commit.

1. The Core Decision Dimensions

Dimension 1: Strategic Differentiation

Does this capability differentiate your product from competitors? Apply the 'newspaper test': if a competitor ships the same capability tomorrow, does your advantage evaporate? Differentiated capabilities should almost always be built — surrendering them to a vendor means surrendering your moat.

- High differentiation (Build): core AI model that drives your product's unique value proposition, proprietary training data that would take competitors years to replicate, domain-specific fine-tuning that delivers measurably better results
- Low differentiation (Buy): infrastructure (authentication, logging, monitoring), commodity capabilities (translation, spell check, basic summarization), capabilities where vendor has 5+ years head start and your use case is standard

Dimension 2: Data Privacy & Regulatory Requirements

This is often the decisive dimension in healthcare, finance, and government. Ask: can vendor receive your data? Under what terms? With what audit rights?

- Data that cannot leave your environment (often Build or self-hosted): PHI/PII without BAA, financial transactions under CCPA/GDPR with strict processing limitations, classified or national security data, data under contractual restriction with source provider
- Data that can go to a vendor (Buy viable): anonymized or synthetic data, publicly available data, data where the vendor has an appropriate BAA or DPA in place

Dimension 3: Capability Fit & Customization Needs

How close is the vendor's off-the-shelf capability to what you need? Measure the gap in specific performance terms, not vibes.

- High fit (Buy): vendor's product scores within 5% of your performance target on your evaluation dataset, integration requires <2 weeks engineering, vendor's roadmap is aligned with your need
- Low fit (Build or partner): vendor product requires significant modification (prompt engineering alone won't close the gap), domain-specific fine-tuning is required, vendor's capability covers 60% of your use case but the 40% gap includes your highest-priority features

Dimension 4: Total Cost of Ownership (3-Year)

Build costs are front-loaded but scale more efficiently. Buy costs are lower initially but compound. The crossover point depends on volume and vendor pricing model.

- Build TCO components: engineering talent (avg. $200K+ fully loaded for senior ML engineer), compute (GPU training + inference), data labeling, evaluation infrastructure, ongoing maintenance (estimate 20-30% of build cost annually), security and compliance tooling
- Buy TCO components: license + usage fees (model API pricing per token / per call), integration engineering, customization limits (what you can't configure you'll rebuild around), vendor lock-in exit cost (migration time + re-training on new system)

2. Scoring Rubric

Score each dimension 1-5. Total score determines recommendation. Adjust weights for your context.

Scoring Guide

- Strategic Differentiation: 5 = core to competitive moat, 3 = useful but replicable, 1 = pure commodity
- Data Sensitivity: 5 = PHI/classified/can't leave perimeter, 3 = PII with contractual controls, 1 = public/non-sensitive
- Capability Gap: 5 = vendor product requires fundamental changes for your use case, 3 = vendor covers 70-80%, 1 = vendor is 95%+ fit
- TCO: 5 = build is cheaper by >20% at 3 years, 3 = within 20%, 1 = buy is >20% cheaper at 3 years
- Team Readiness: 5 = have the talent + infra to build, 3 = could build with 1-2 hires, 1 = no ML capability

Decision Thresholds

- Score 20-25: Strong Build signal — pursue internal development
- Score 13-19: Neutral — run a detailed cost model and vendor pilot before deciding
- Score 5-12: Strong Buy signal — vendor is the right answer

Override conditions: any single dimension scoring 5 on Data Sensitivity is a hard Build/self-host regardless of total score. Any dimension scoring 1 on Team Readiness combined with tight timeline is a hard Buy.

3. Decision Tree

Gate 1: Can your data leave your environment?

NO -> Build or self-hosted vendor (open-source model in your cloud). YES -> continue to Gate 2.

Gate 2: Is this capability core to your competitive differentiation?

YES -> Build (unless timeline is <3 months AND vendor meets 90%+ of requirements). NO -> continue to Gate 3.

Gate 3: Does a vendor product meet 80%+ of your performance requirements today?

NO -> Build (or partner with vendor to customize). YES -> continue to Gate 4.

Gate 4: Is the 3-year TCO for Buy within 30% of Build?

NO (Buy is >30% more expensive at scale) -> Build. YES -> Buy, with vendor lock-in mitigation strategy.

Gate 5: Do you have the team to build and maintain this?

NO -> Buy (or Partner, which means Buy + dedicated technical collaboration with vendor). YES -> make final decision on score and strategic factors.

4. Cost Model Templates

Build Cost Model Template

- Year 1 — Build: (Engineers x $220K fully loaded) + ($50K compute for training) + ($30K data/annotation) + ($40K tooling) = ~$340K for 1 engineer + infra
- Year 2 — Maintenance & Iteration: 40% of Year 1 build cost + ($0.05-0.20/query inference cost at volume) + incremental data cost
- Year 3 — Scale: Inference cost dominates; engineering cost if maintaining/improving; compliance audit cost if regulated domain

Buy Cost Model Template

- OpenAI API: GPT-4o $2.50/1M input tokens, $10/1M output tokens. At 10M calls/month with avg 500 input + 200 output tokens: $1,250 input + $2,000 output = $3,250/month = $39K/year. Scale to 100M calls: $390K/year.
- Anthropic API: Claude Sonnet $3/1M input + $15/1M output. Similar volume: ~$4,800/month, $57K/year.
- Azure OpenAI / AWS Bedrock: 10-30% cheaper with committed spend, adds enterprise SLAs and BAA availability.
- Integration cost: 2-4 weeks engineering = $20-40K one-time + $10-20K/year maintenance

Break-Even Analysis

General heuristics: if annual API cost exceeds $200K, internal inference often becomes cost-competitive with open-source models. If you have the engineering talent, self-hosting Llama-3-70B or Mistral on dedicated GPU instances typically runs at $0.001-0.003/call — 5-10x cheaper than frontier model APIs at scale.

5. Vendor Evaluation Process

Evaluation Checklist

- [ ] Performance: run vendor's product on your evaluation dataset (not theirs). Score against your defined thresholds.
- [ ] Security & Compliance: SOC 2 Type II report available? BAA available (for healthcare)? DPA for GDPR? Data retention policy? Data used for model training?
- [ ] SLA & Reliability: uptime SLA? P99 latency? Rate limits? What happens when you hit rate limits? Disaster recovery?
- [ ] Pricing Model Stability: is pricing locked via contract or can vendor change it with 30 days notice? What's the pricing trajectory (check if vendor has raised prices in last 2 years)?
- [ ] Lock-in Risk: can you export your data and models? If vendor shuts down or raises prices 5x, what is migration timeline and cost?
- [ ] Roadmap Alignment: does vendor's 12-month roadmap address your 12-month needs? Who is their target customer — are you aligned or an edge case?

Recommended Pilot Structure

2-week technical pilot before any contract. Pilot must include: (1) performance benchmark on your actual data, (2) integration prototype with your tech stack, (3) security review of data flows, (4) cost projection based on actual usage patterns during pilot. Never sign a multi-year contract without a completed technical pilot.

6. Risk Assessment Framework

Build Risks

- Talent risk: ML engineers are expensive and scarce. What happens if your key ML engineer leaves? Document model architecture and training procedures from day one.
- Performance risk: building something good enough to ship takes 3-6x longer than estimated. Apply a 2x multiplier to any build timeline estimate in your planning.
- Opportunity cost risk: every week spent on infrastructure is a week not spent on features. Quantify what you're not building.

Buy Risks

- Vendor lock-in: switching vendor means re-engineering integrations, potentially re-annotating training data, and often losing fine-tuning investments. Mitigation: abstract the AI layer behind an internal interface so swapping vendors requires changing one module, not the entire product.
- Pricing risk: API-first AI vendors have rapidly changed pricing. Anthropic, OpenAI, and Cohere have both increased and decreased prices dramatically. Build in contract protections: locked pricing for committed spend, pricing change notice requirements.
- Quality drift risk: vendor model updates can silently change output behavior. Mitigation: pin model version where possible, run automated regression tests after every vendor model update, monitor output distribution in production.
- Regulatory risk: if vendor is acquired, raises prices prohibitively, or fails, you may need to rebuild on a different foundation. Ensure your regulatory clearances (if applicable) can transfer to a different underlying model.

7. Migration Planning

Vendor Exit Plan (Should Build on Day 1)

Every buy decision should include a vendor exit plan written on day 1. This is not pessimism — it is risk management. The plan should cover: (1) data portability — can you export all training data, evaluation data, and model artifacts? (2) interface abstraction — is your code isolated from the vendor SDK or embedded throughout? (3) alternative vendors — identify 2 backup vendors and estimate migration effort, (4) self-host option — identify an open-source model that could replace vendor capability, even at 10% quality reduction.
The act of writing the exit plan often reveals lock-in risks that change the build vs. buy decision. If you can't write a credible exit plan, the vendor lock-in risk is very high.