
LLM Adoption Exercise: FAQs and Answers

General & Framework

Q1. What is the purpose of this in-class exercise?

To simulate CIO decision-making for LLM adoption. Teams of 4–5 role-play as technology leaders for three firm types (small/mid/large), complete an Adoption Decision Table, select a 2×2 quadrant, recommend models, define governance, and design a 90-day pilot.

Q2. What are the four quadrants in the 2×2 adoption matrix?

The exercise forces teams to pick a quadrant *before* naming any model:

- **Q1** — Plug-and-Play API (closed, external, turnkey SaaS)
- **Q2** — Open-Source-as-a-Service (hosted open-weights via Together AI, Replicate, Hugging Face)
- **Q3** — Private Cloud / Managed Private (closed frontier on gov/enterprise cloud, e.g., Bedrock, Azure OpenAI with BAA)
- **Q4** — Full Private Stack / Self-Hosted (open-weights on your own GPU cluster, air-gapped if needed)

Q3. Why does the exercise say "state the quadrant before the model"?

Because architecture precedes vendor. If you pick a model first, you lock into that vendor's constraints. Picking a quadrant first forces you to match deployment mode to data class, compliance needs, volume, and talent — then choose a model that fits.

Team A — Small Firm (Healthcare)

Q4. Why did Team A choose Q1 (Plug-and-Play API) instead of self-hosting?

12-person clinic, no ML talent, low volume (~ 5K queries/mo). Self-hosting vLLM would require GPU ops, SRE, and pipelines they don't have. Q1 gives fastest ROI with zero infrastructure.

Q5. Why Claude Haiku 4.5 specifically?

Haiku is the lowest-cost, fastest Anthropic model — sufficient for clinical-protocol lookup and billing FAQs. Delivered via AWS Bedrock, it supports HIPAA compliance with a BAA and zero-retention endpoint.

Q6. What is Team A's biggest risk and how is it mitigated? Data leakage via prompts.

Mitigation: zero-retention endpoint, automatic PII/PHI redaction, and a strict rule — no patient names in queries. Also requires a DPA + BAA with the vendor.

Q7. What trap did Team A avoid?

Buying a GPU server for "privacy" with no pipelines, no talent, and no maintenance plan — an

expensive shelf ornament.

Team B — Mid-Market (Financial Services)

Q8. Why does Team B use a Q2→Q4 migration path?

Start with Q2 (hosted open-weights like Llama 4 on Together AI/Replicate) to move fast, then migrate high-volume internal RAG to Q4 (self-hosted) as volume and compliance demands grow. This avoids upfront GPU capital while building toward full control.

Q9. What is Team B's routing split and why?

- ~80% → Llama 4 Maverick (self-hosted, high-volume extraction/classification)
- ~15% → DeepSeek V4 Flash (commodity classification)
- <5% → Claude Sonnet 5 (complex disputes, compliance explanations)

Routing is by **task difficulty**, not price alone. Simple tasks get cheap models; complex tasks get flagship reasoning.

Q10. What is Team B's top risk and mitigation?

RAG ACL leakage — retrieval returning documents across permission groups (e.g., one loan officer seeing another's files). Mitigation: role-based access control (RBAC), document-level permissions, eval harness, and circuit breakers.

Q11. Why does Team B need a VPC?

GLBA and PCI-DSS require that PII stays within a controlled network boundary. A VPC keeps inference traffic off the public internet, satisfies Safeguards Rule requirements, and reduces egress exposure.

Team C — Large Regulated (Public Sector)

Q12. Why does Team C use Q3 + Q4 instead of Q1 or Q2?

Federal agency with 12,000 staff, FedRAMP High, ITAR, GDPR, and classified-adjacent data. Q1/Q2 mean data leaves government infrastructure or relies on commercial multi-tenant clouds — unacceptable. Q3 handles public-tier chatbot via FedRAMP-approved private cloud; Q4 handles internal policy RAG on self-hosted open-weights.

Q13. Why mix GPT-6 Sol, Qwen3-27B, and Phi-4?

- **GPT-6 Sol** (<1%) — FedRAMP-authorized frontier for rare complex public-facing queries
- **Qwen3-27B** (~85%) — MIT-licensed self-hosted for internal RAG (cost-effective at 50M+ tokens/month)
- **Phi-4** (~14%) — edge/offline deployment for air-gapped scenarios

Q14. What is the geopolitical risk with Qwen and how is it mitigated?

Qwen is developed by a PRC-based company (Alibaba). Risk: supply-chain/weight provenance, potential backdoor, US regulatory exposure. Mitigation: self-host only on air-gapped US GovCloud, weight provenance audit, no PRC-cloud egress, board-level geo-risk review.

Q15. What governance layers does Team C implement?

Model gateway → eval harness → human-in-loop (for disputes) → rollback policy → FinOps dashboard. This creates defense-in-depth from request routing to cost monitoring.

Key Concepts

Q16. What do API, DPA, and BAA mean in this context?

- **API** — Application Programming Interface; the technical channel through which your app sends prompts and receives completions.
- **DPA** — Data Processing Agreement; a contract requiring vendors to handle your data only for agreed purposes, with deletion/security obligations (required under GDPR/CCPA).
- **BAA** — Business Associate Agreement; a HIPAA-mandated contract for any vendor handling PHI on behalf of a covered entity.

Q17. What is the difference between PII, PHI, and PCI-DSS data?

- **PII** — Personally Identifiable Information (name, SSN, email, etc.)
- **PHI** — Protected Health Information (PII + health context, held by a HIPAA-covered entity)
- **PCI-DSS** — Security standard for payment card data (credit card numbers, CVV) Each maps to different regulations: GLBA (financial), HIPAA (health), PCI-DSS (payments).

Q18. What is a model-agnostic gateway and why does the exercise recommend it?

A routing layer that sits between your application and multiple LLM backends. It lets you swap models (when the leaderboard flips) without rewriting application code. Combined with an eval harness, it enables routing by task difficulty and protects against vendor lock-in.

Strategy & Takeaways

Q19. What does "route by task difficulty, not price alone" mean?

Don't send every request to the most expensive flagship model. Use cheap/small models (Flash, Haiku) for classification and extraction (~80% of traffic), reserve mid-tier for drafting/RAG (~15%), and flagship only for complex reasoning (<5%). This optimizes both cost and quality.

Q20. Why quantify 5-year TCO instead of 1-month API sticker price?

1-month API cost looks cheap but ignores: (a) volume growth making API cumulative cost exceed self-hosting, (b) SRE/talent costs for self-hosted, (c) compliance/audit overhead, (d) egress fees, (e) vendor price changes. 5-year TCO reveals the true economic picture.

Q21. What is the "digital-divide constraint" mentioned in the debrief?

The capital threshold (GPU cluster, talent, SRE) required to move from Q1/Q2 (rented API/hosted) to Q3/Q4 (private/self-hosted). Small firms often can't cross it alone; solutions include external funding, managed open endpoints, or cooperative infrastructure.

Q22. How should the 90-day pilot be structured?

As a business experiment with a baseline and a go/no-go decision, not a technology demo. Define success metrics upfront (accuracy, cost/query, adoption rate, zero leakage), measure against baseline, and decide whether to scale, pivot, or kill the project.

Final Thought (from the exercise)

- *Do not bet on one model.*
- *Build a model-agnostic gateway with an eval harness and intelligent routing.*
- *When the leaderboard flips next quarter — and it will — you swap a model, not your architecture.*