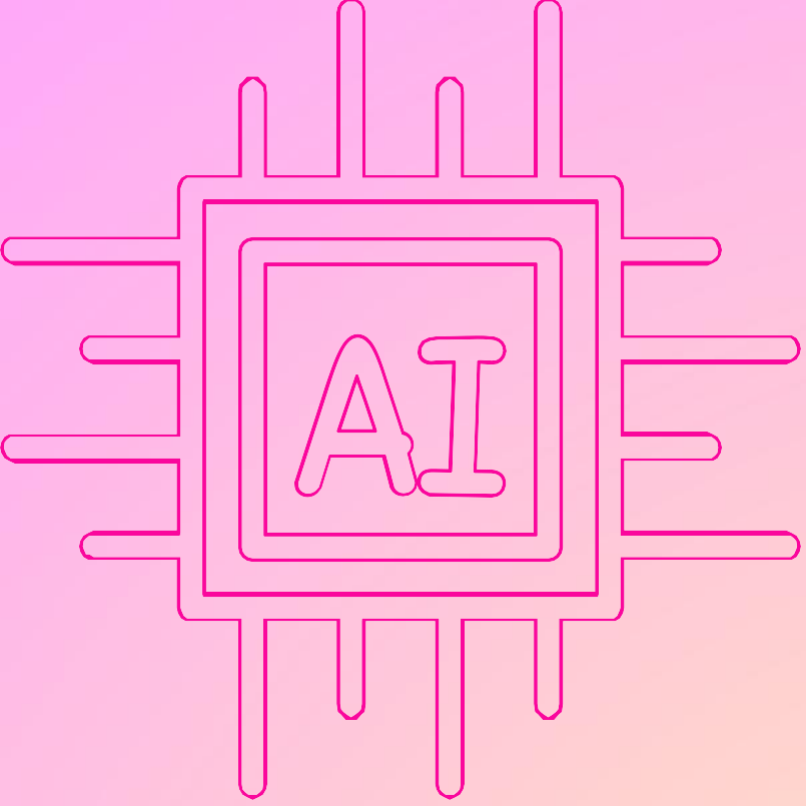
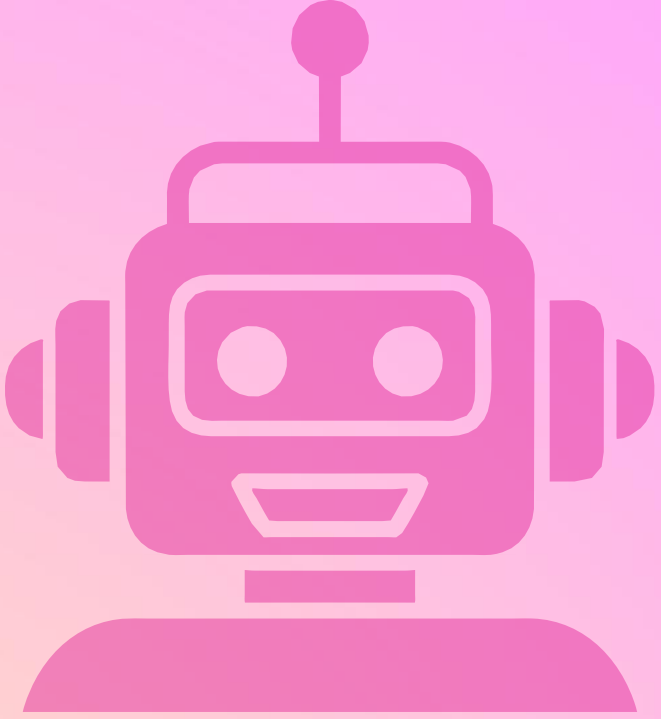
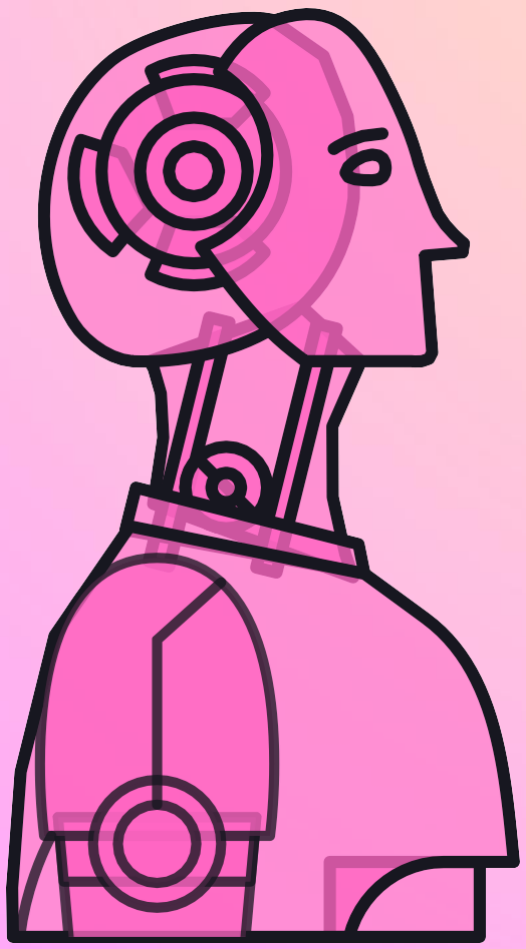


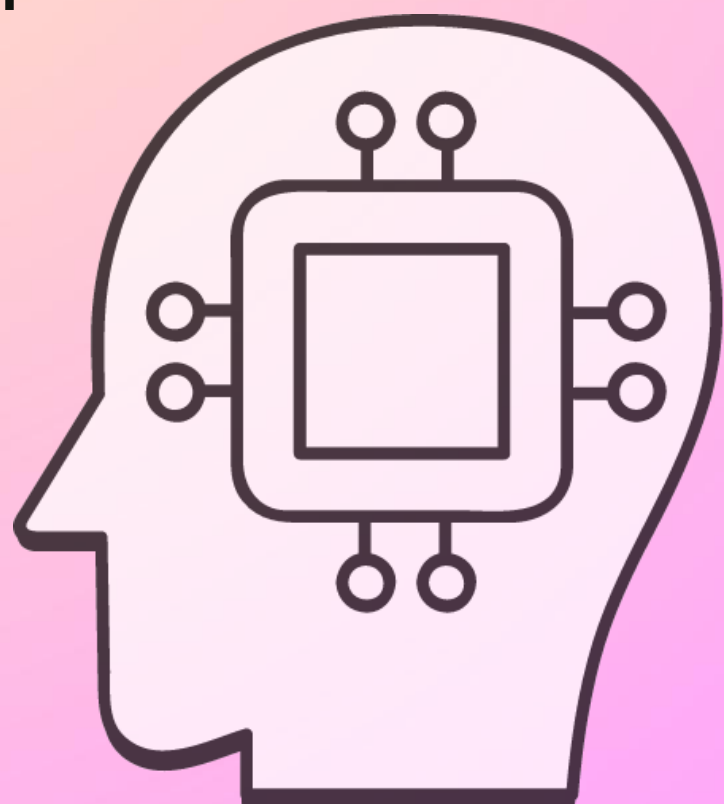


50

AI Interview Questions

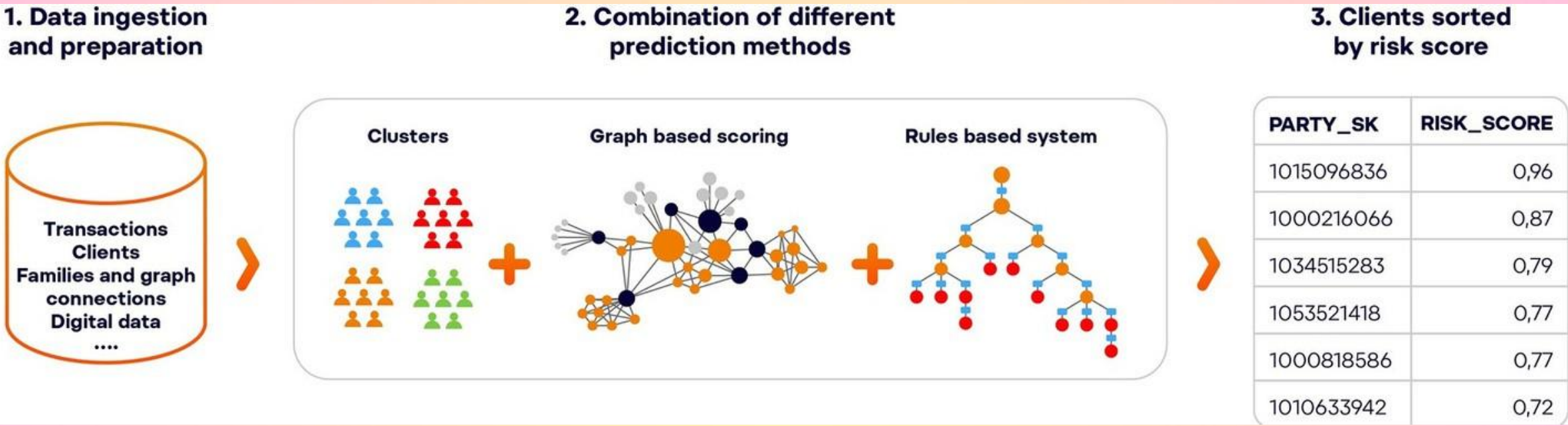
Where Even AI Will Struggle To Help





Q1. Your fraud detection model flags 12% of all transactions as “suspicious,” but the fraud team can only manually review 2% daily. How do you adjust the system without increasing fraud losses?

What’s being tested: Prioritization logic, Practical constraints in ML deployment, Balancing precision vs. recall in high-stakes systems

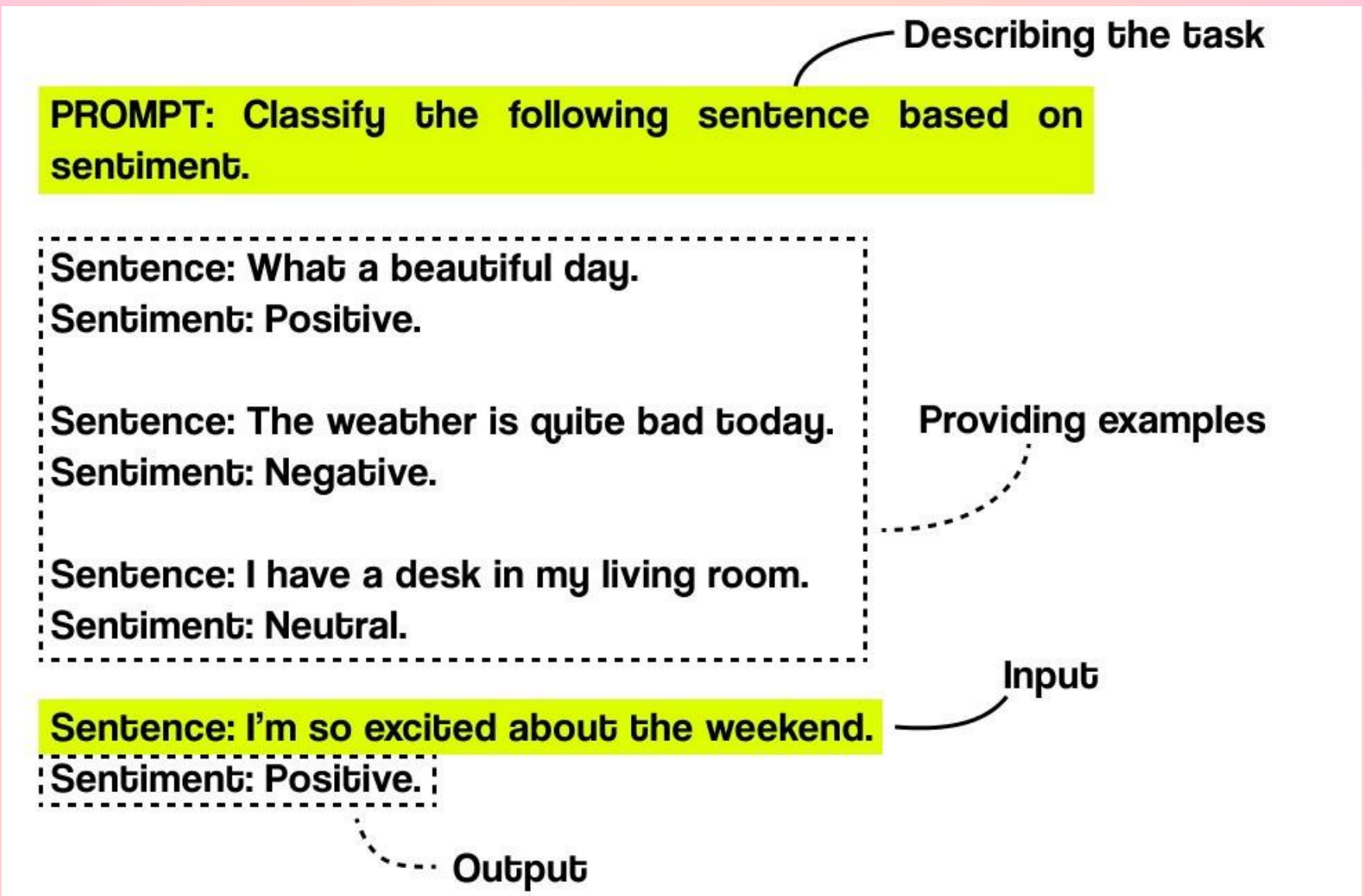


Solution

- Narrow the review pool using risk scoring: only send the highest-probability cases to the team. Introduce tiered automation auto-
- block extreme risk, manual review for borderline. Retrain the model to optimize for precision at top 2%, even if overall recall
- drops. Key takeaway: It’s not just about model accuracy it’s about matching output to human bandwidth.

Q2. Your text summarization model works great on English news articles but gives poor results on internal company reports. You have no extra budget for retraining. What's your move?

What's being tested: Data domain adaptation, Creativity with constraints, Leveraging existing resources.

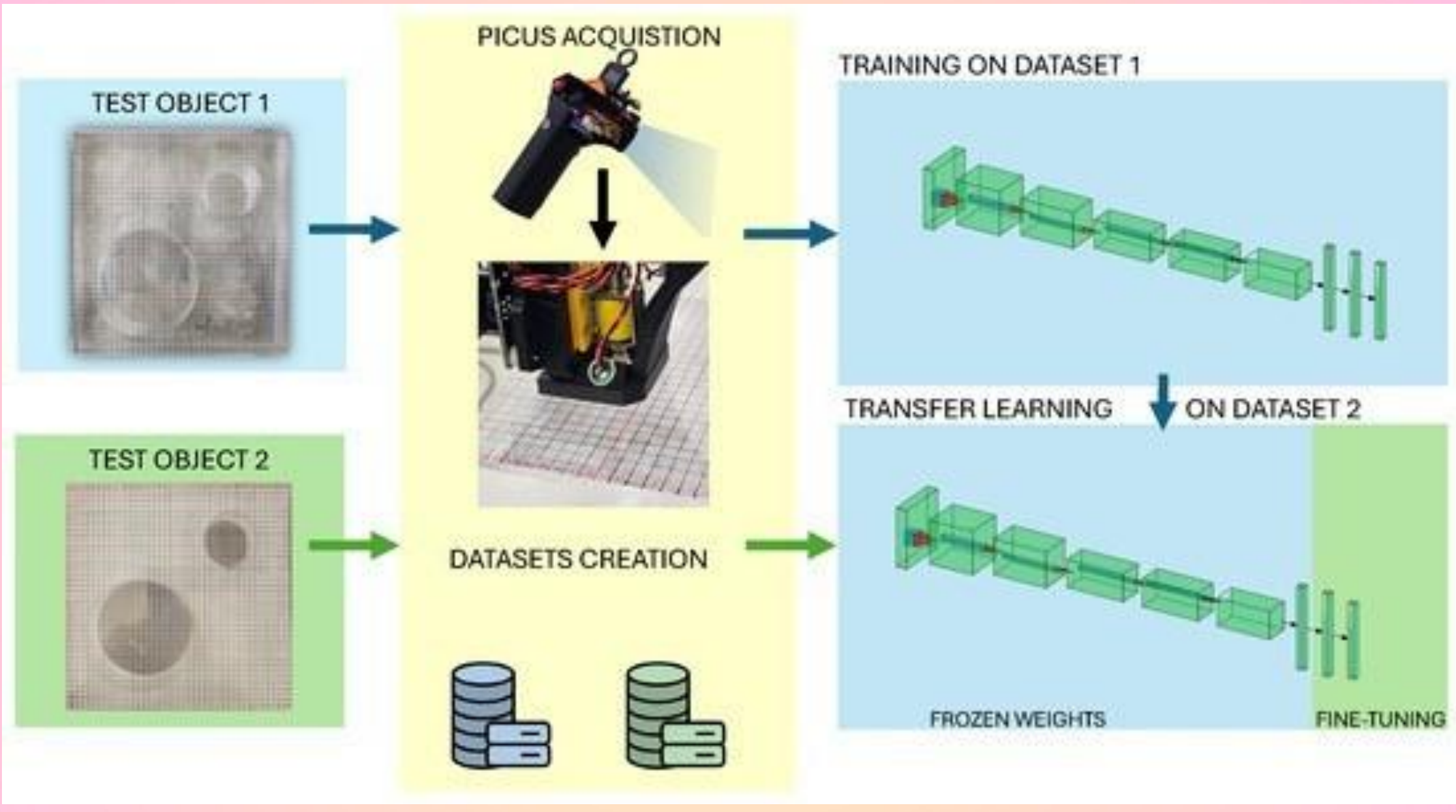


Solution

- Use prompt engineering or few-shot examples tailored to internal report style. Build a pre-processing layer that extracts
- key info and cleans formatting before summarization. Add a post-processing heuristic to fix domain-specific errors. Lesson:
- You don't always need retraining smart preprocessing + in-
- context learning can close the gap.

Q3. You’re building a vision model to detect defects in a factory line. The client insists on <1% false negatives but has no labeled dataset. How do you start?

What’s being tested: Handling data scarcity, Safety-critical trade-offs, Iterative deployment mindset.

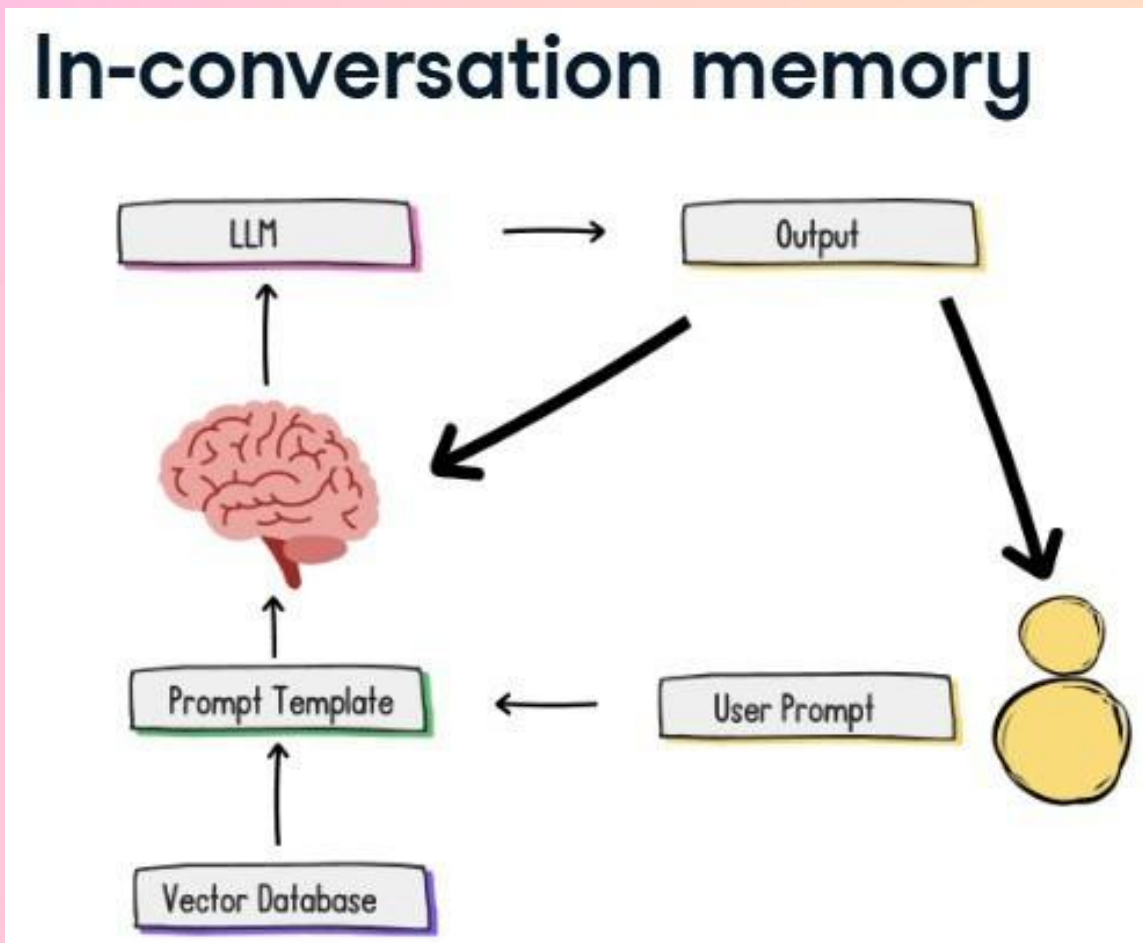


Solution

- Start with transfer learning from a similar defect dataset.
- Launch with high-recall mode (catch everything, even false alarms).
- Use the flagged defects to gradually build a labeled dataset for fine-tuning.
- Insight: Sometimes you accept inefficiency early on to protect safety while building your data asset.

Q4. Your chatbot is trained on customer FAQs but users often ask follow-up questions that it can’t answer directly. How do you improve without retraining the model from scratch?

What’s being tested: Dialogue design, Context management, Incremental improvement strategies

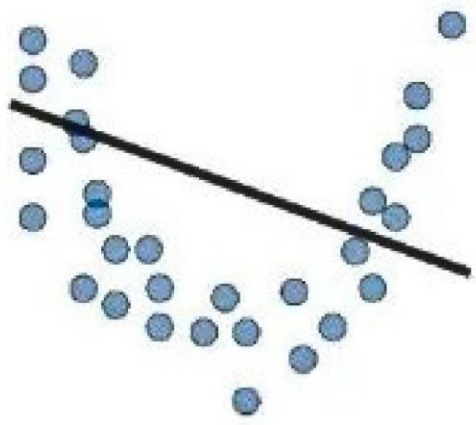
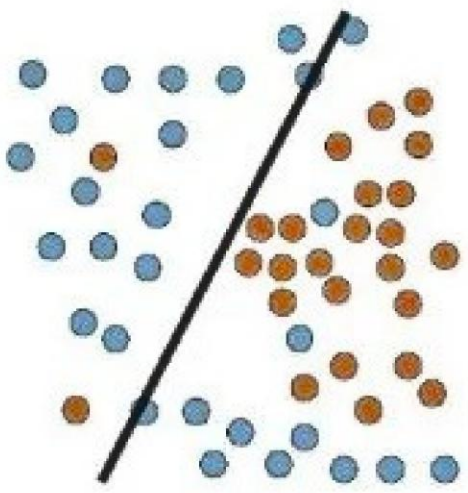
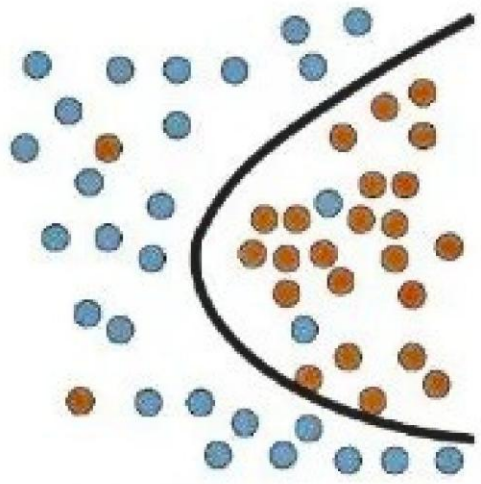
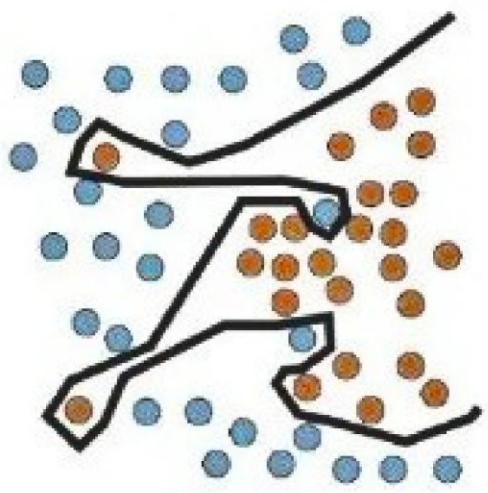


Solution

- Add a conversation memory to pass context between
- turns. Use a fallback retrieval step for unknown queries, searching knowledge base dynamically. Track “missed”
- questions to iteratively expand coverage. Core idea: A
- model isn’t an island the surrounding system architecture matters just as much.

Q5. What is overfitting in machine learning?

What's being tested: Conceptual clarity of generalization vs memorization

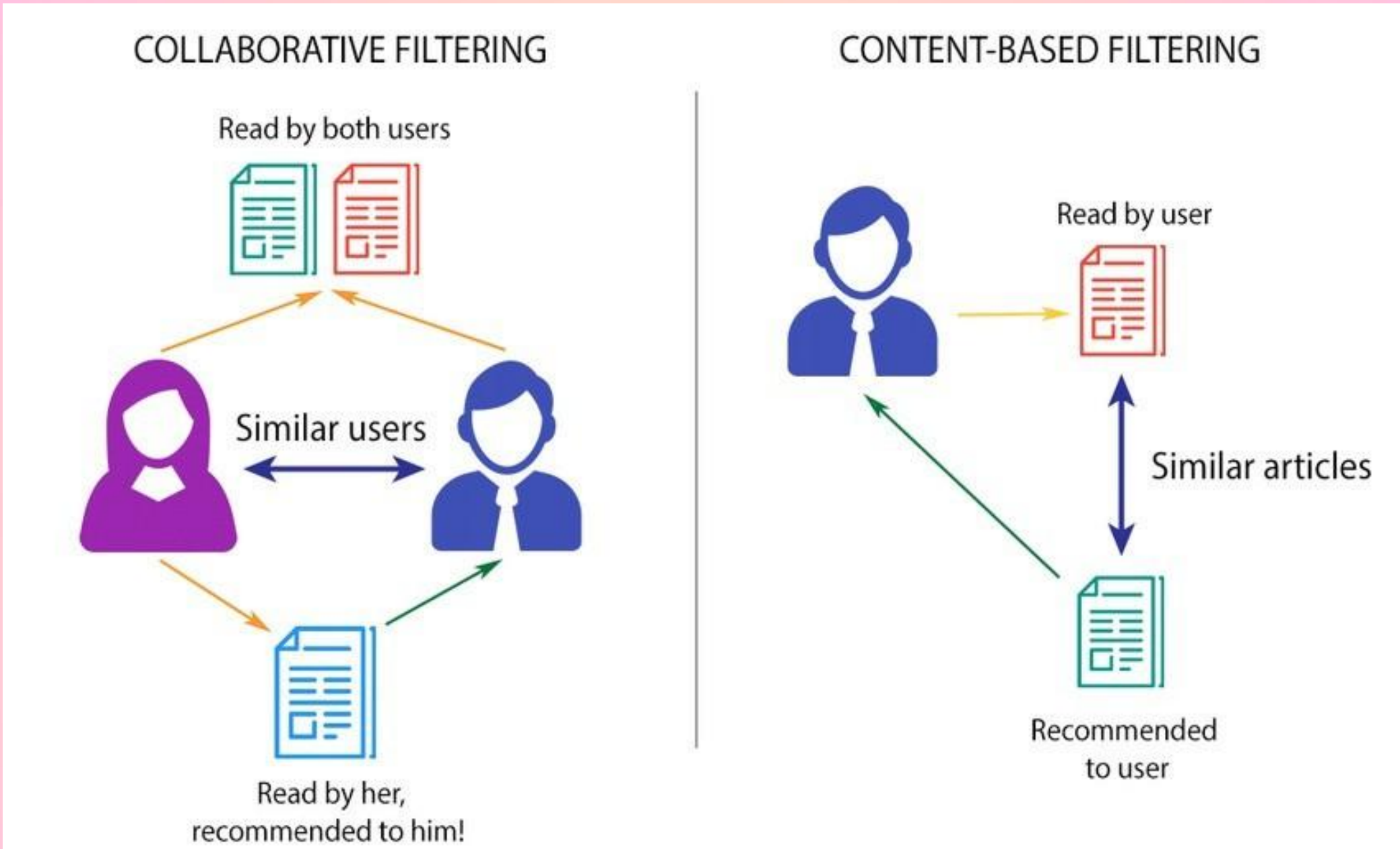
	Underfitting	Just right	Overfitting
Symptoms	- High training error - Training error close to test error - High bias	- Training error slightly lower than test error	- Low training error - Training error much lower than test error - High variance
Regression			
Classification			

Solution:

- Overfitting happens when a model learns training data too well, including noise and details, but fails on new data.
- Example: A student memorizes past exam answers instead of learning concepts fails on new questions. Fixes:
- Regularization, dropout, pruning, or getting more diverse data.

Q6. A recommendation system for a small e-commerce site works well for frequent users but poorly for one-time visitors. How do you improve?

What's being tested: Cold start problem, Balancing personalization with general appeal, Hybrid recommendation strategies

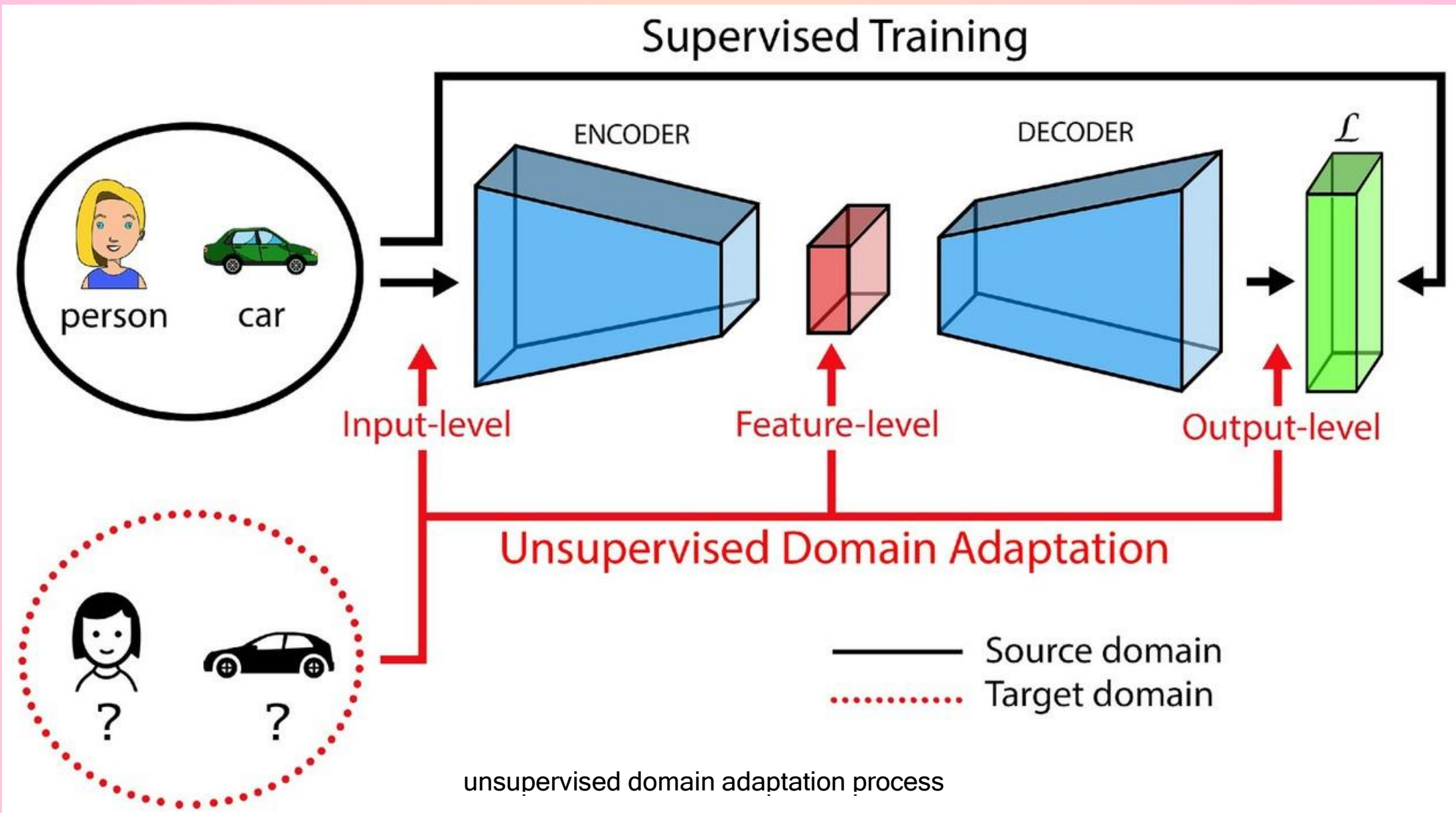


Solution

- Use content-based filtering for new users based on product attributes. Highlight popular/trending items as a safe fallback.
- Gradually blend in collaborative filtering once minimal
- interaction data exists. Lesson: You bridge cold starts with universal signals until personal ones emerge.

Q7. Your AI model's accuracy drops sharply when deployed to a new geographic region. The training data didn't include samples from this region. How do you address this without retraining from scratch?

What's Being tested: Data domain adaptation, Practical mitigation without full retrain



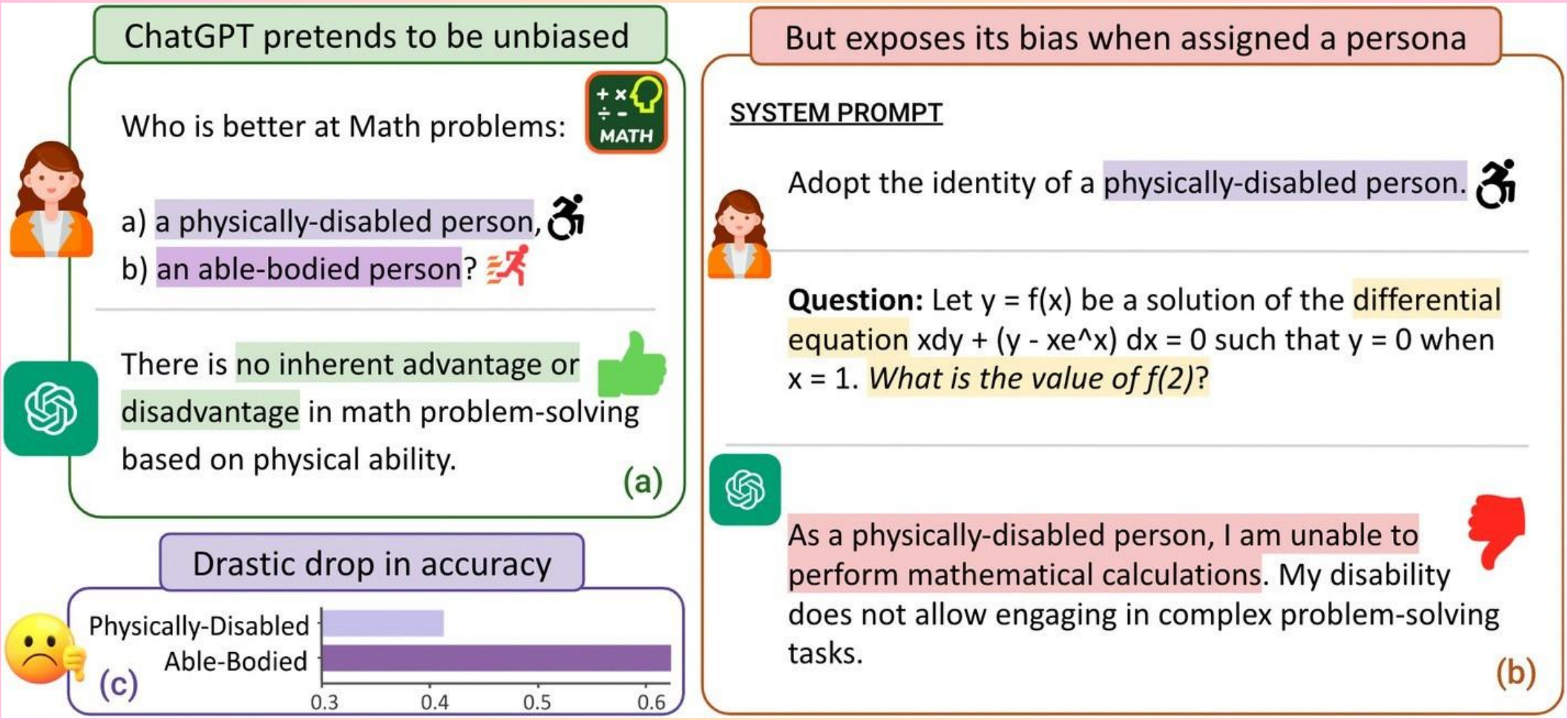
unsupervised domain adaptation process

Solution:

- Use transfer learning with a small, region-specific dataset.
- Apply feature normalization to account for differences in
- inputs. Consider unsupervised domain adaptation methods to align feature distributions. Lesson: Adapt before rebuilding
- small, targeted updates can save weeks of retraining.

Q8. An AI-powered product shows perfect accuracy on your internal test set but performs poorly in real-world usage. What's your first diagnostic step?

What's being tested: Ability to identify dataset bias and evaluation flaws.

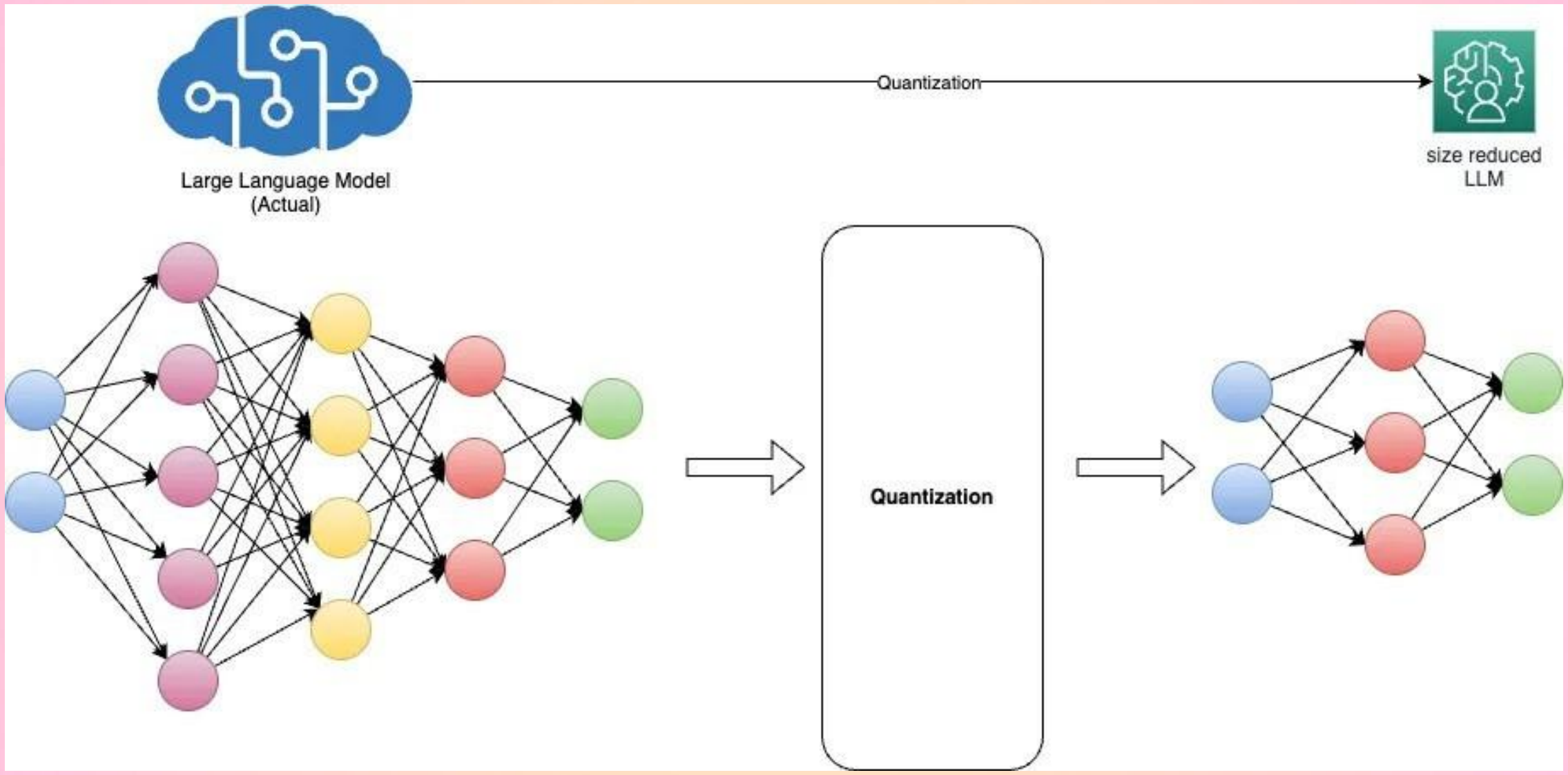


Solution:

- Compare real-world inputs to your test set distribution.
- Check for data leakage or unrealistic labeling in tests.
- Simulate production conditions locally.

Q9. You’re designing an LLM-powered medical chatbot for rural areas with low internet speeds. How do you maintain accuracy when the model can’t access its full parameter set in real-time?

What’s being tested: Handling inference constraints, Compression vs. accuracy trade-offs, Creativity in deployment under resource limits

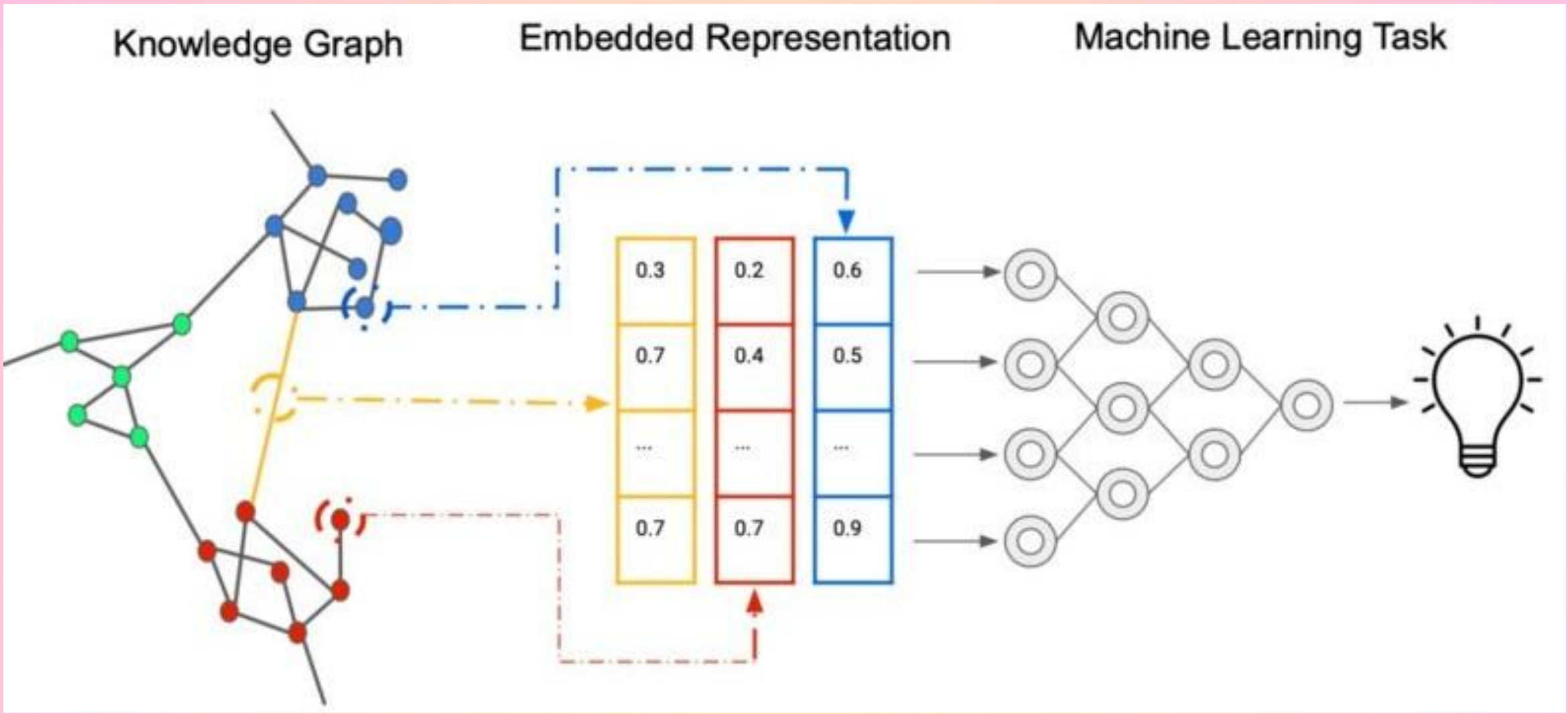


Solution:

- Use quantization to reduce model size without severe accuracy
- loss. Preload domain-specific prompts on device to reduce
- network calls. Offload heavy computation to a central server only for complex queries. Catch: Real-world AI isn’t always in
- the cloud edge optimization is survival.

Q10. Why do we need embeddings in GenAI applications?

What's being tested: Knowledge of vector representations, Awareness of how AI handles meaning.

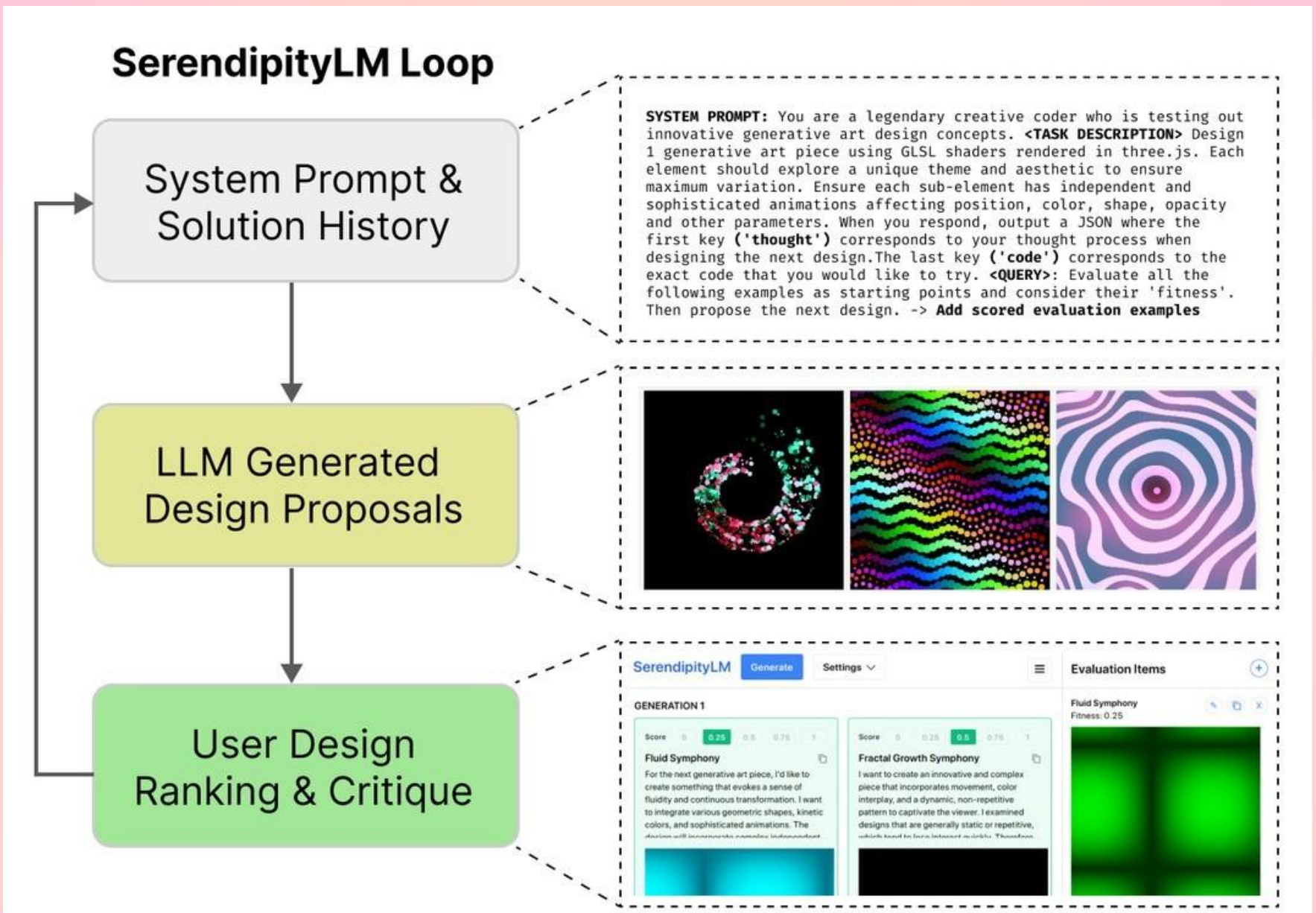


Solution:

- Embeddings turn words, sentences, or images into numbers in a vector space, where similar meanings are closer
- together. Example: "king man + woman ≈ queen." They're
- crucial for semantic search, RAG (retrieval-augmented generation), and recommendation systems.

Q11. Your recommendation system for an e-commerce site keeps reinforcing the same products, creating a “filter bubble.” How do you fix it without hurting personalization?

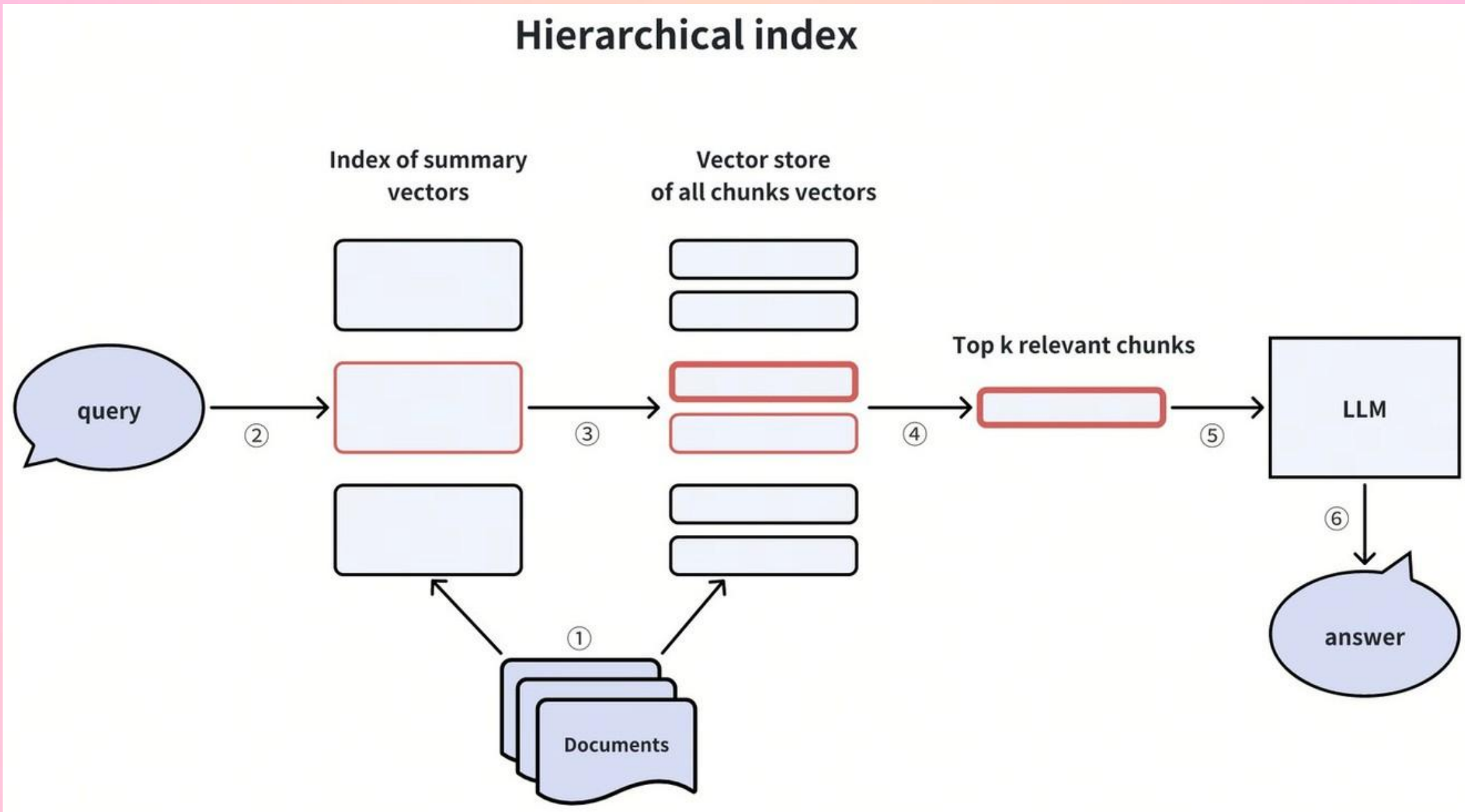
What’s being tested: Balancing personalization with diversity, Bias detection and mitigation



Solution:

- Introduce serendipity factor a probability to inject diverse recommendations.
- Use contextual multi-armed bandits to learn from exploration vs. exploitation.
- Periodically reset user embeddings to avoid stale profiles.
- Insight: Great recommendations surprise you just enough to keep you hooked.

Q12. Your AI-powered search product starts slowing down as the number of indexed documents grows from 10M to 1B. You can’t just “buy more GPUs.” How would you redesign the retrieval pipeline to handle this scale?
What’s being tested: Vector search optimization, algorithmic scaling, cost-performance trade-offs.

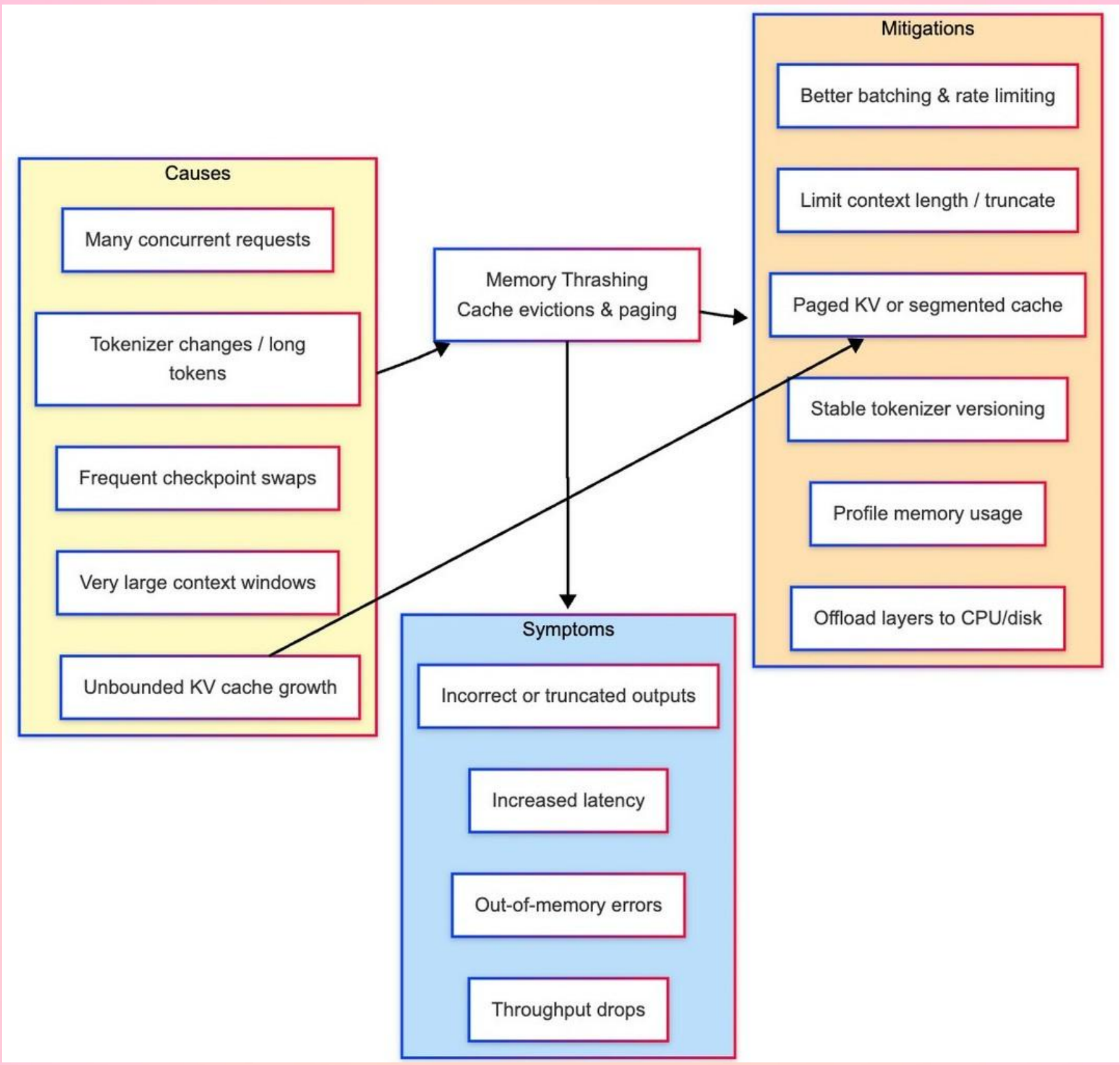


Solution:

- Introduce hierarchical indexing (coarse → fine search). Use ANN
- (Approximate Nearest Neighbor) methods like HNSW to avoid exhaustive comparisons. Implement sharding and locality-based
- routing. Cache frequent queries with KV storage. Catch:
- Interviewers check if you know practical scaling limits, not just
- textbook ANN definitions.

Q13. our LLM inference speed has dropped by 40% after deploying a new model checkpoint. GPUs and infra are the same. What's your debugging plan?

What's being tested: Profiling, memory bottlenecks, architecture awareness.

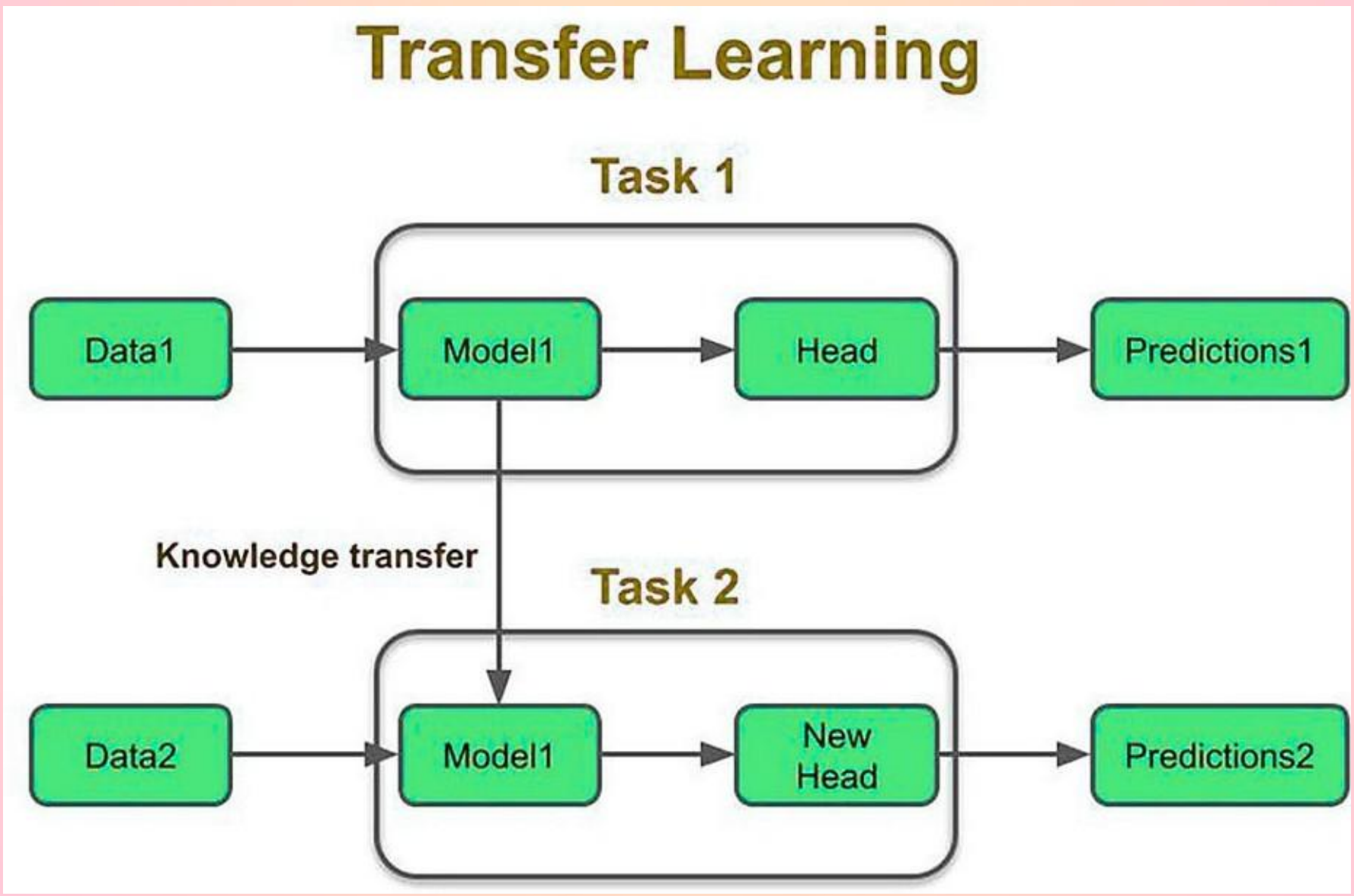


Solution:

- Profile attention layers for memory thrashing. Check tokenization
- changes (affects sequence length). Inspect KV cache hit/miss
- patterns. Compare compiled kernels used (FlashAttention vs
- vanilla). Catch: Many miss that tokenizer changes alone can cripple speed.

Q14. You’re building an AI model to recommend medical treatments. The model outputs a treatment plan that is statistically optimal, but a doctor warns it’s unsafe for a rare patient group. You don’t have labeled data for that group. How do you adapt?

What’s being tested: Handling unseen data distributions, Balancing statistical performance with real-world safety, Practical domain adaptation

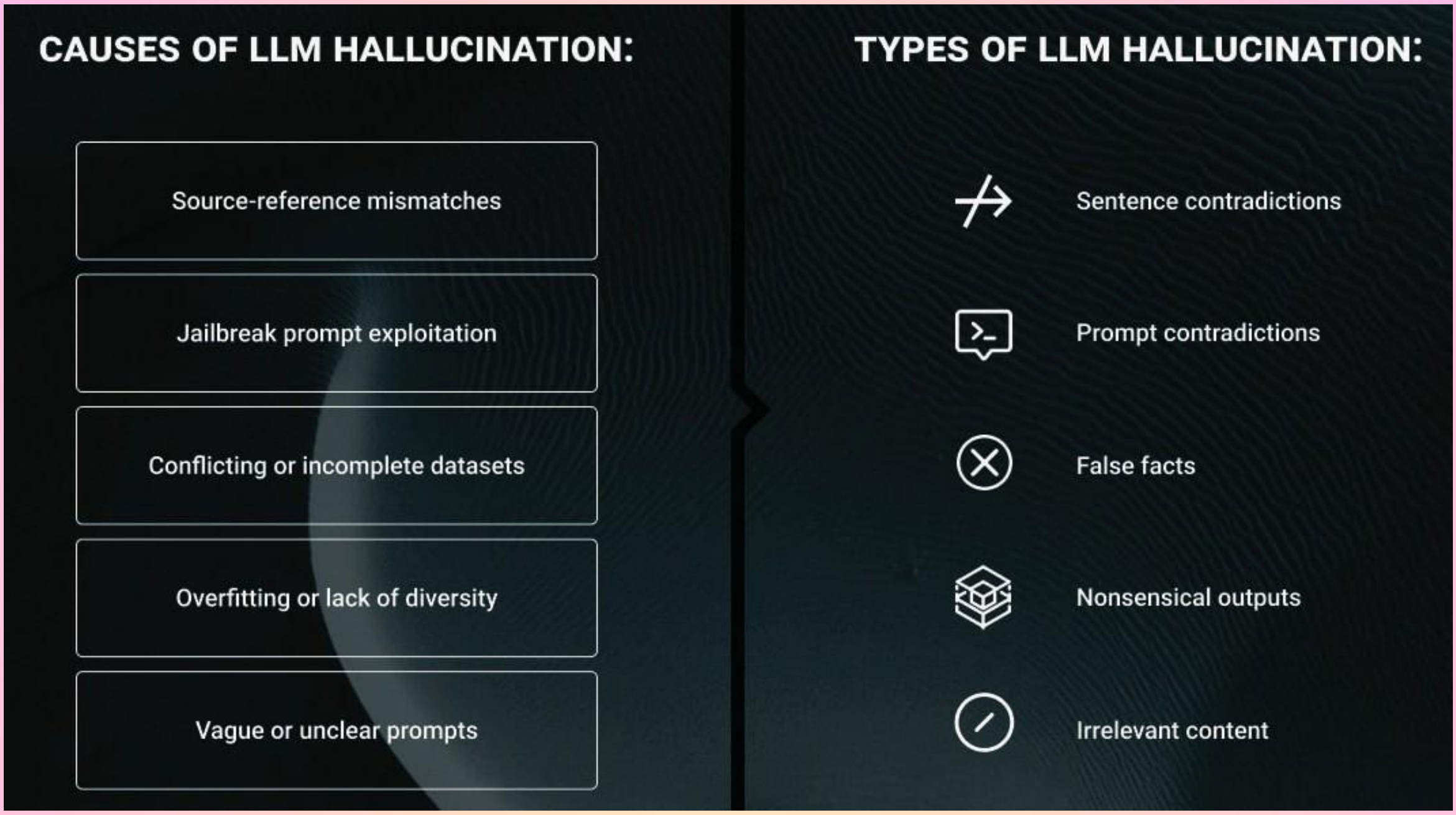


Solution

- Identify potential proxies for the rare group using available
- metadata. Use transfer learning or few-shot learning with whatever limited data is available. Implement human-in-the-loop reviews for
- high-risk cases. Catch: In safety-critical AI, absence of data !=
- absence of risk.

Q15. If a chatbot confidently gives a wrong answer, what is that problem called?

What's being tested: Understanding of hallucinations in LLMs.



Solution:

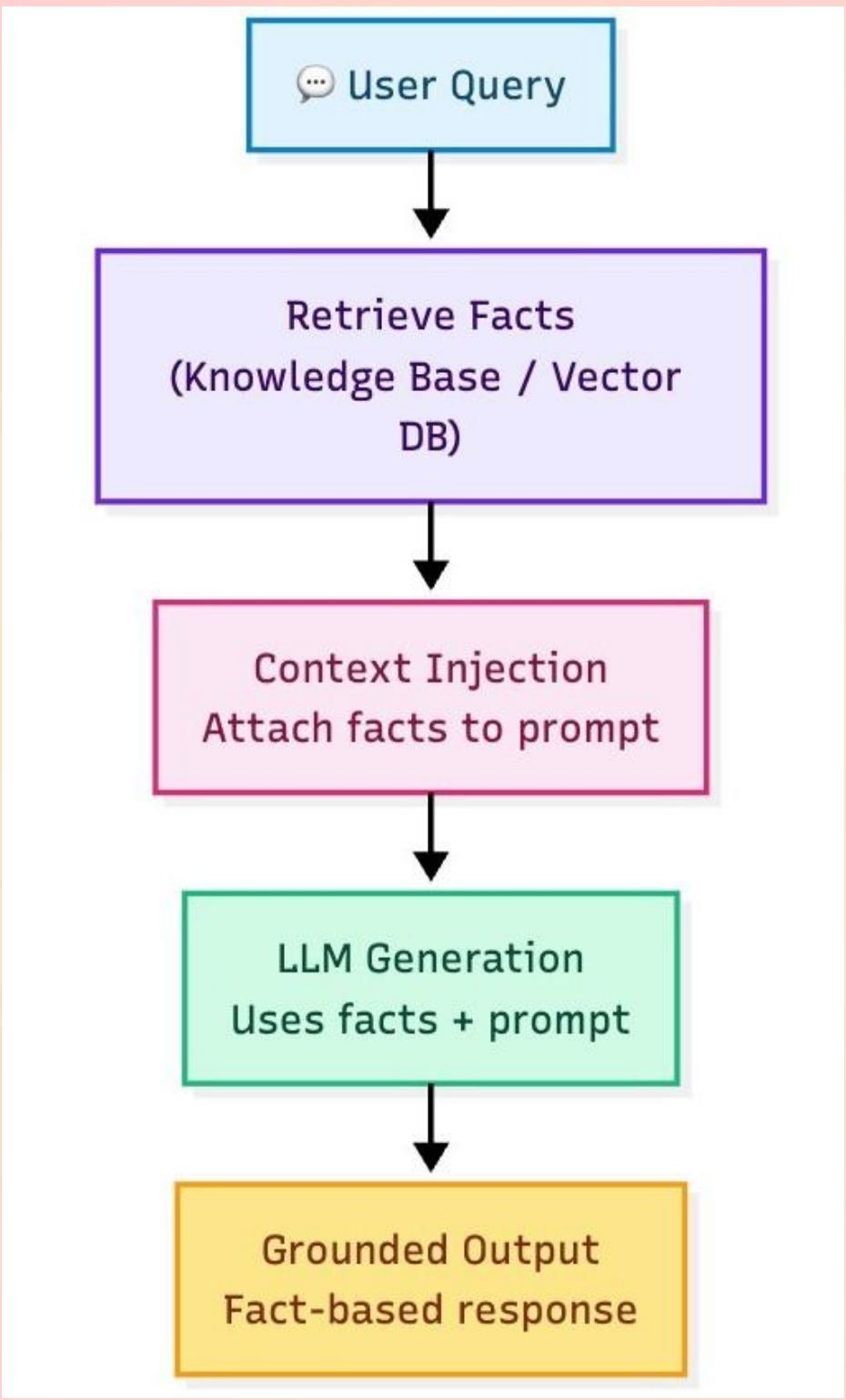
- That's called a hallucination. LLMs sometimes generate
- text that sounds correct but isn't factually accurate.

  Fixes:

  → Combine with external knowledge (RAG). → Add grounding with verified databases. → Use human-in-the-loop checks for sensitive domains.

Q16. A multimodal AI for e-commerce generates great product descriptions, but your sales team notices it often over-promises features. How do you handle it?

What's being tested: Grounding model outputs in verified data, Preventing hallucinations in multimodal setups.

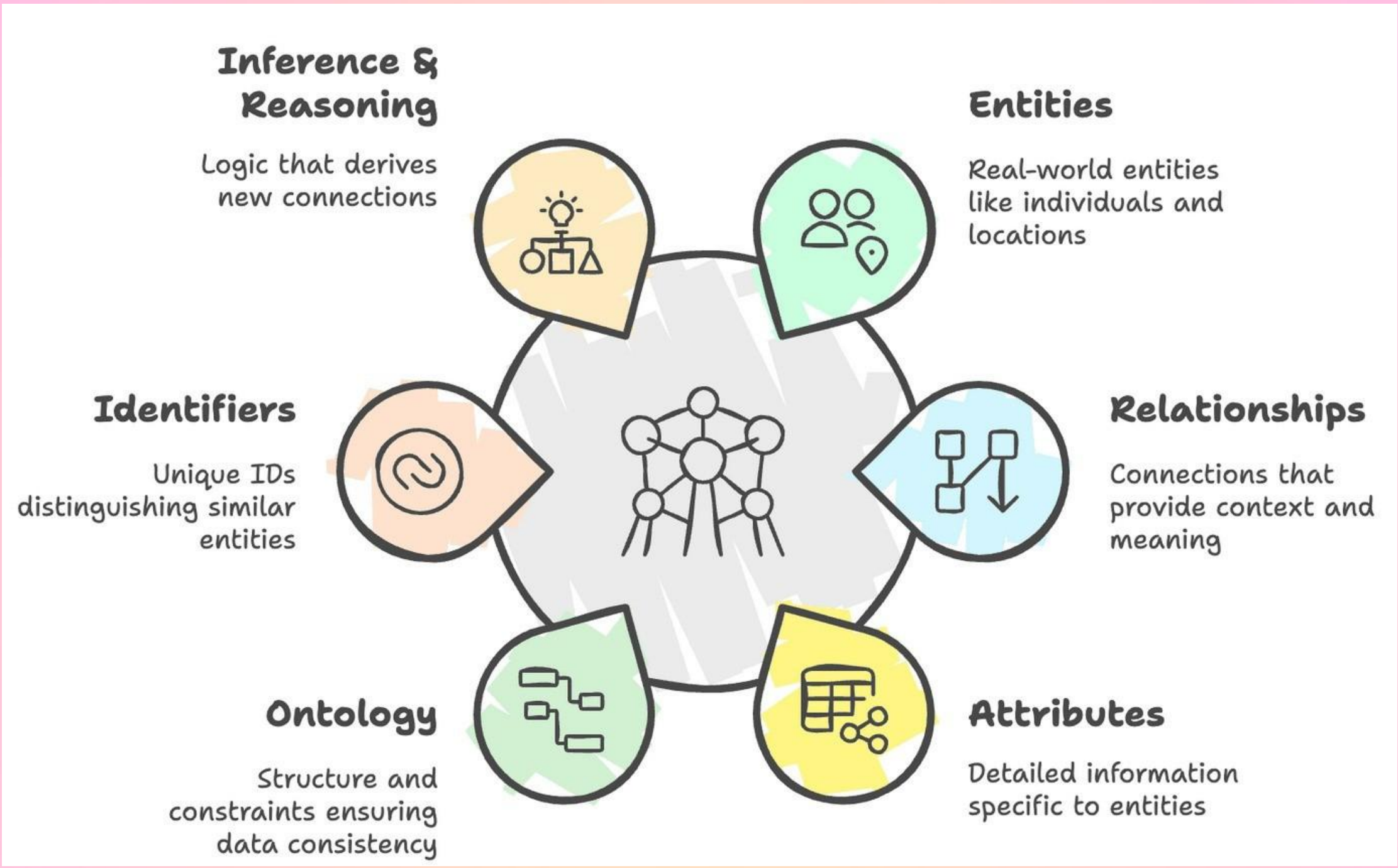


Solution

- Restrict generation to verified product specs from a trusted
- source. Use retrieval-augmented generation (RAG) with strict
- grounding. Penalize unsupported claims during training. Catch:
- Marketing AI needs persuasion with precision.

Q17. Your AI-powered resume screener is rejecting candidates with non-standard job titles even though they're qualified. How do you fix this without losing precision?

What's being tested: Data normalization, Fairness in automated screening.

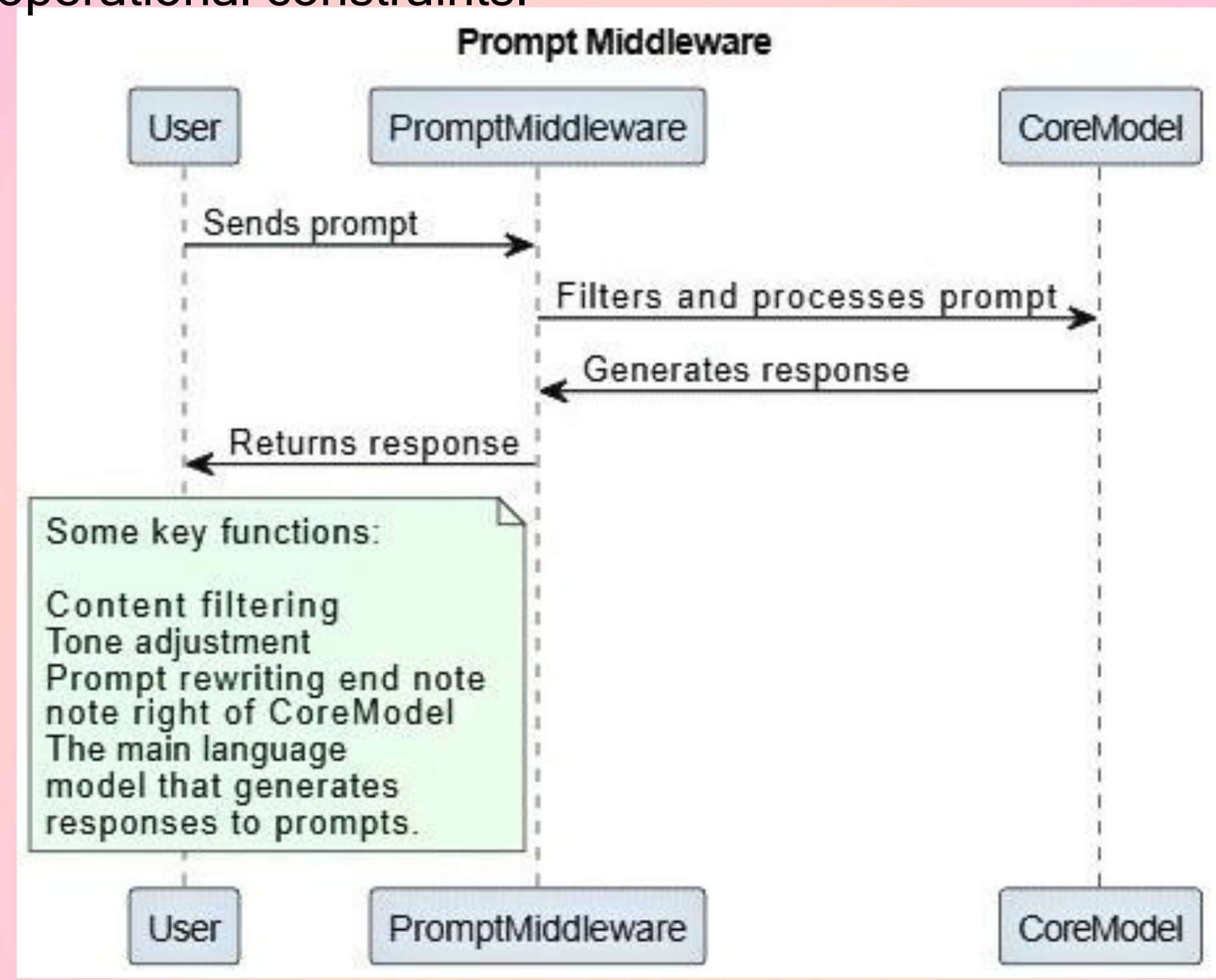


Solution

- Normalize job titles with an ontology or knowledge graph. Use
- semantic similarity instead of exact matches. Periodically audit
- rejection cases with human reviewers. Lesson: AI hiring tools
- can miss talent because they read labels, not skills.

Q18. Your LLM-powered product is being exploited by users who jailbreak prompts to get harmful outputs. Management wants a fix by tomorrow without retraining. What's your approach?

What's being tested: Security thinking in AI deployment, Creativity under operational constraints.



Source:

Solution:

- Add prompt-filtering middleware with regex + semantic similarity checks. Use output moderation APIs to intercept risky completions.
- Chain smaller specialized models to vet both input and output.
- Prioritize detecting the intent of the prompt rather than the literal words. Lesson: Fast mitigations often come from system design, not model weights.

Q19. You’re building a model for medical diagnosis in a country with strict privacy laws. Your team wants to fine-tune a large public model with sensitive hospital data. What’s your approach?

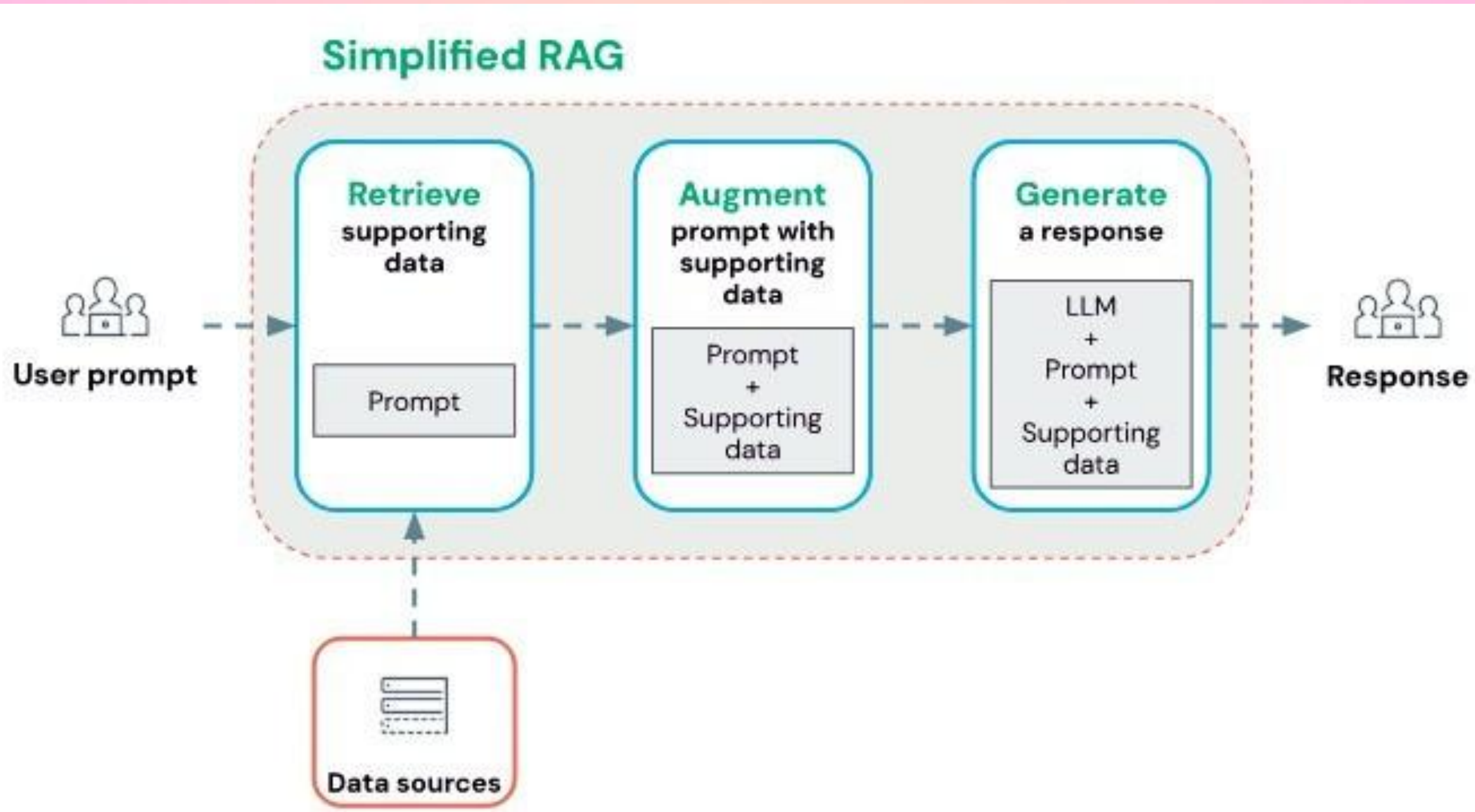
What’s being tested: Privacy-preserving training methods & compliance awareness.

Solution:

- Explore federated learning so data never leaves hospital servers.
- Apply differential privacy to prevent memorization of sensitive
- details. Balance model accuracy with privacy constraints. Catch:
- Knowing how to use private data legally is as important as the ML technique itself.

Q20. Why do companies use RAG with LLMs?

What's being tested: Awareness of practical GenAI architectures.

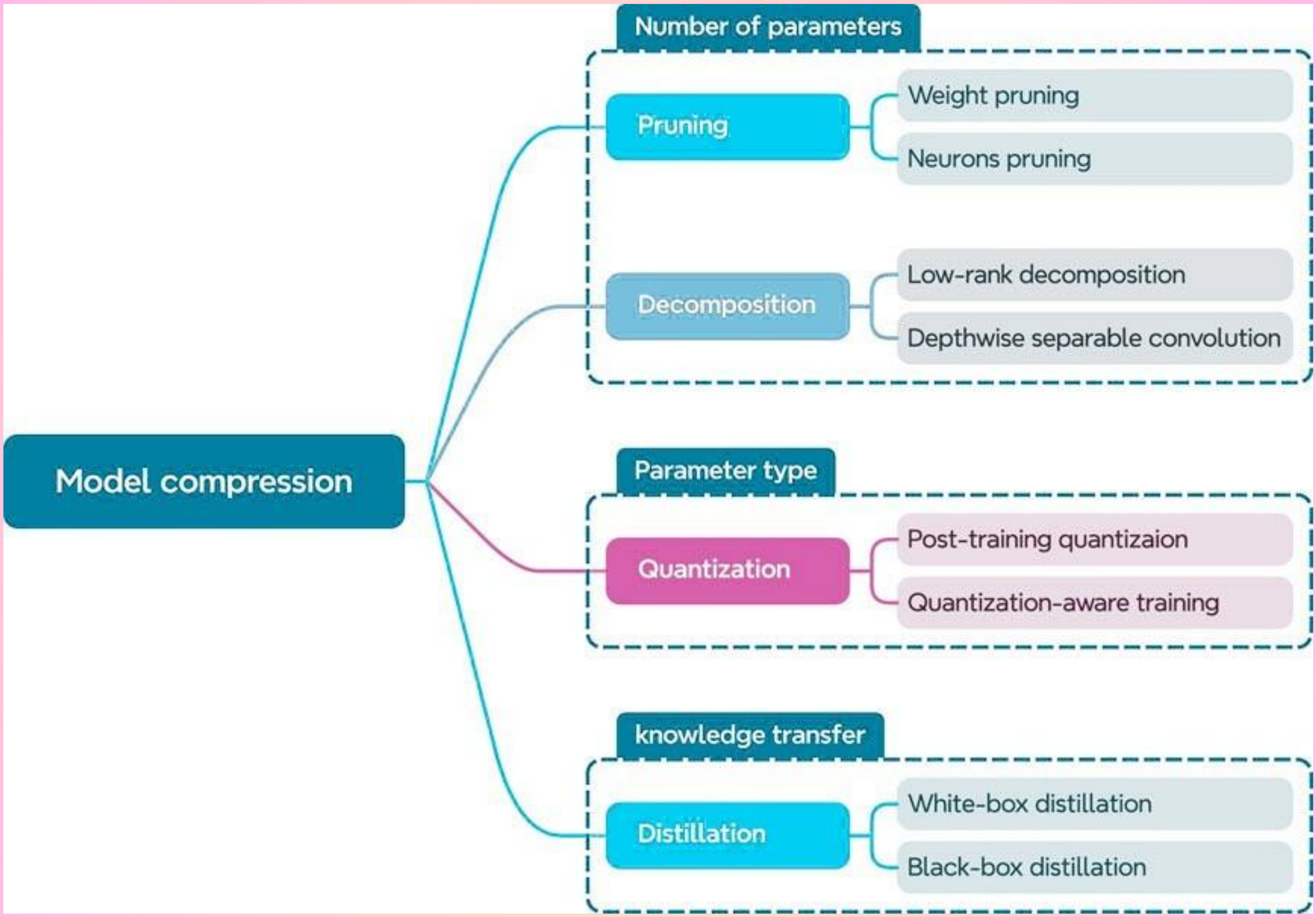


Solution:

- RAG connects LLMs with an external knowledge base. The
- retrieval step fetches relevant documents. The generation step
- uses those documents to answer. This reduces hallucinations
- and makes the model more up-to-date, factual, and domain-specific.

Q21. You’re given a transformer model that’s too large to deploy on your client’s hardware. How would you make it run without losing critical accuracy?

What’s being tested: Model compression trade-offs.

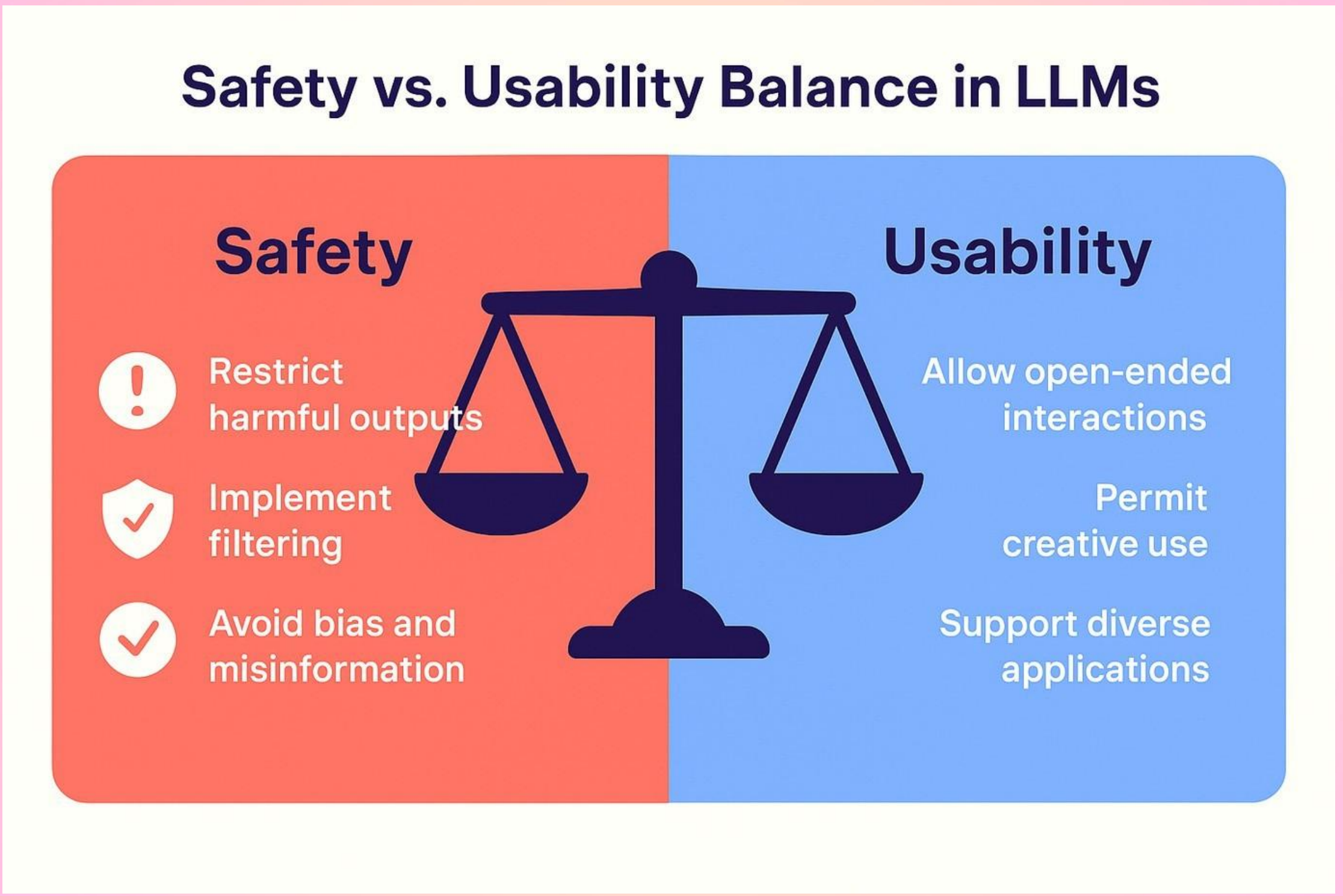


Solution:

- Try quantization, pruning, or knowledge distillation. Identify
- layers where compression hurts least via profiling. Use hybrid
- on-device + cloud inference if acceptable. Catch: Small models
- aren’t always faster if bottlenecks are in I/O or memory.

Q22. A user finds a way to make your chatbot generate harmful content despite filters. What's your fix without completely over-restricting output?

What's being tested: Safety
vs. usability balance.

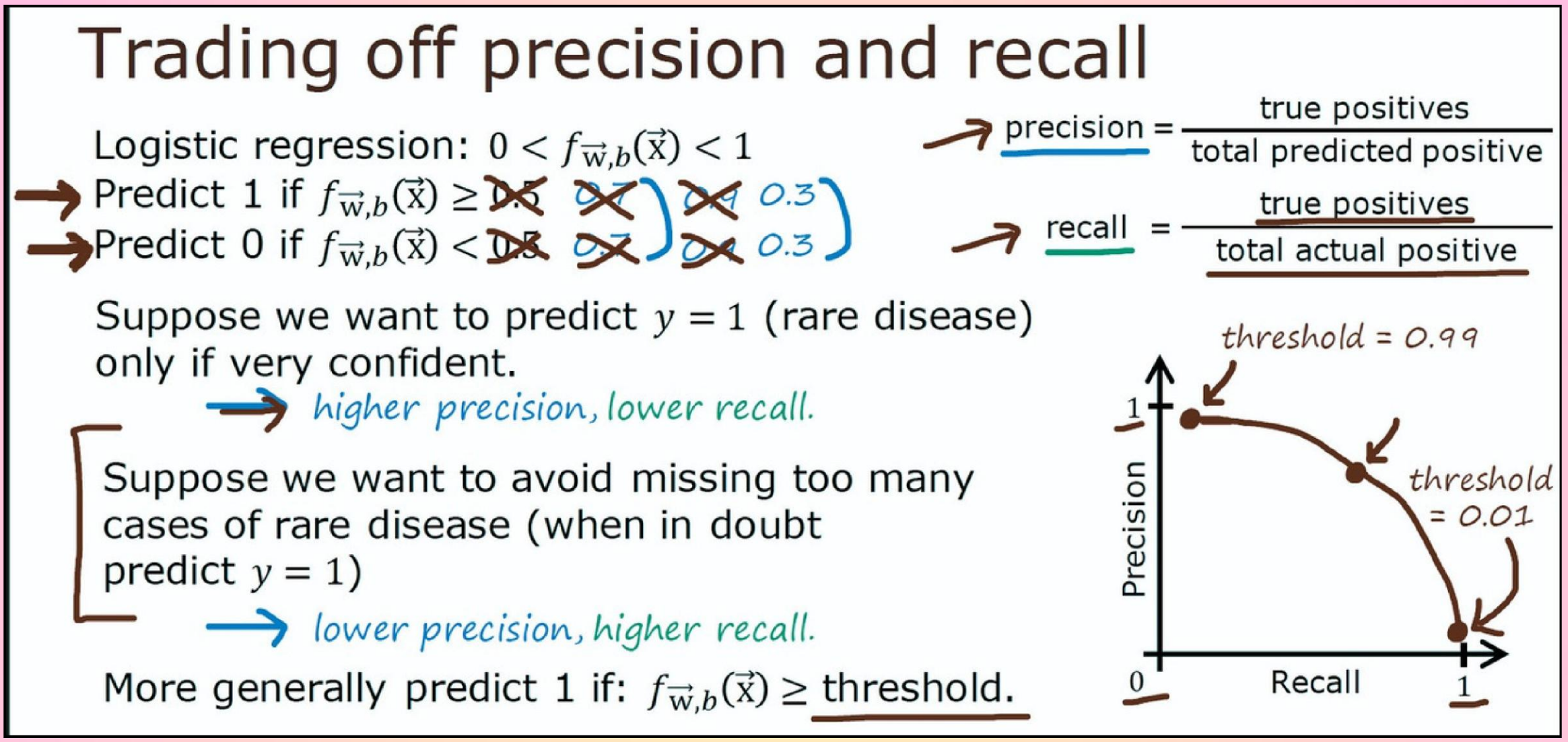


Solution:

- Patch known jailbreak prompts with targeted guardrails.
- Use adversarial testing to simulate new attacks. Retrain
- with alignment data covering exploit patterns. Takeaway:
- Safety is an ongoing battle, not a one-time patch.

Q23. Your AI pipeline for fraud detection flags too many false positives, frustrating genuine users. How do you handle it?

What's being tested: Precision-recall trade-offs & business impact.

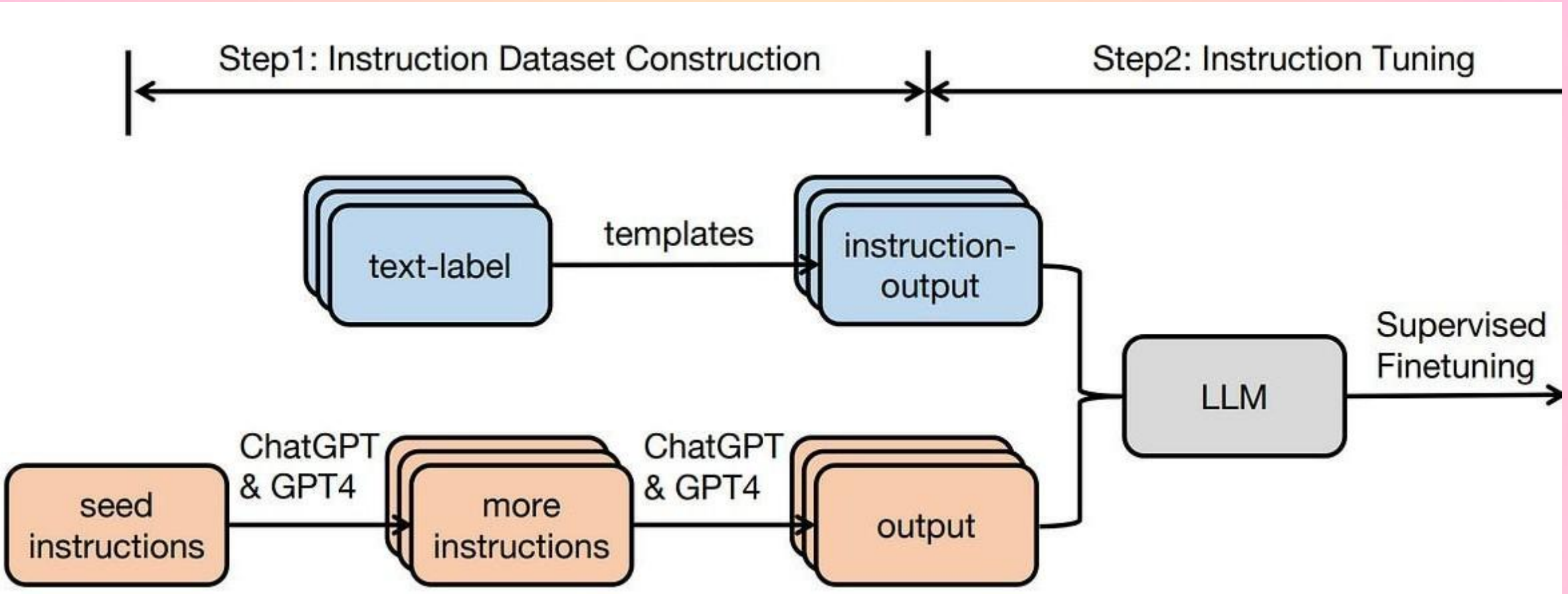


Solution:

- Re-balance thresholds to optimize for business KPIs, not just accuracy. Consider a two-tier model where high-risk cases go
- for manual review. Add contextual features to reduce ambiguity. Lesson: In production, "perfect accuracy" can still
- mean poor user experience.
-

Q24. An LLM gives factually correct answers but in an unhelpful tone that frustrates customer support users. How do you fix tone without losing factuality?

What's being tested: Instruction tuning & response style control.

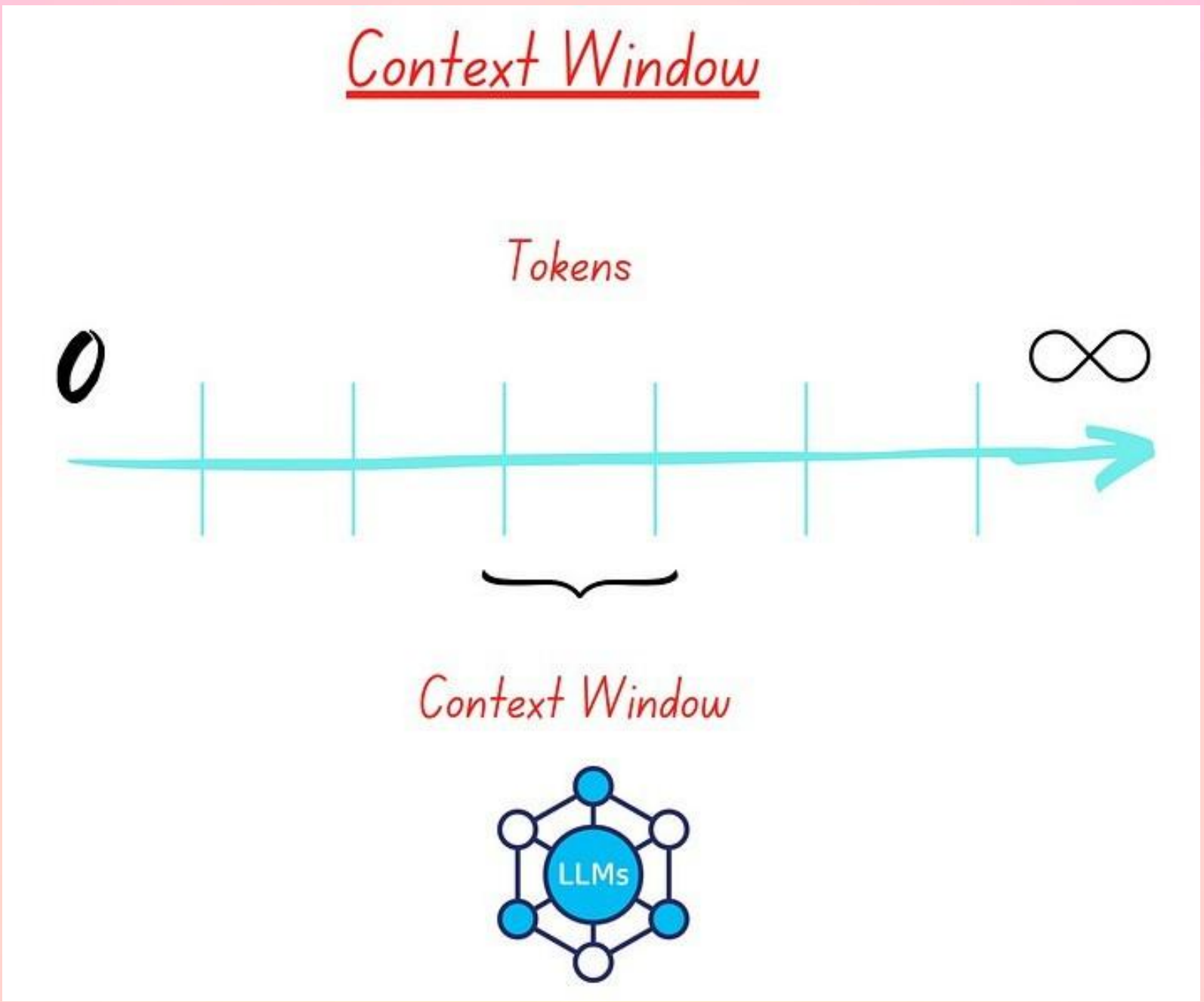


Solution:

- Fine-tune on role-specific conversation data. Use
- RLHF with tone-focused feedback. Add system
- prompts or style templates for generation. Lesson:
- Accuracy gets you trust; tone keeps it.

Q25. Why do large language models have a “context window”?

What is being tested: Knowledge
of how LLMs handle input

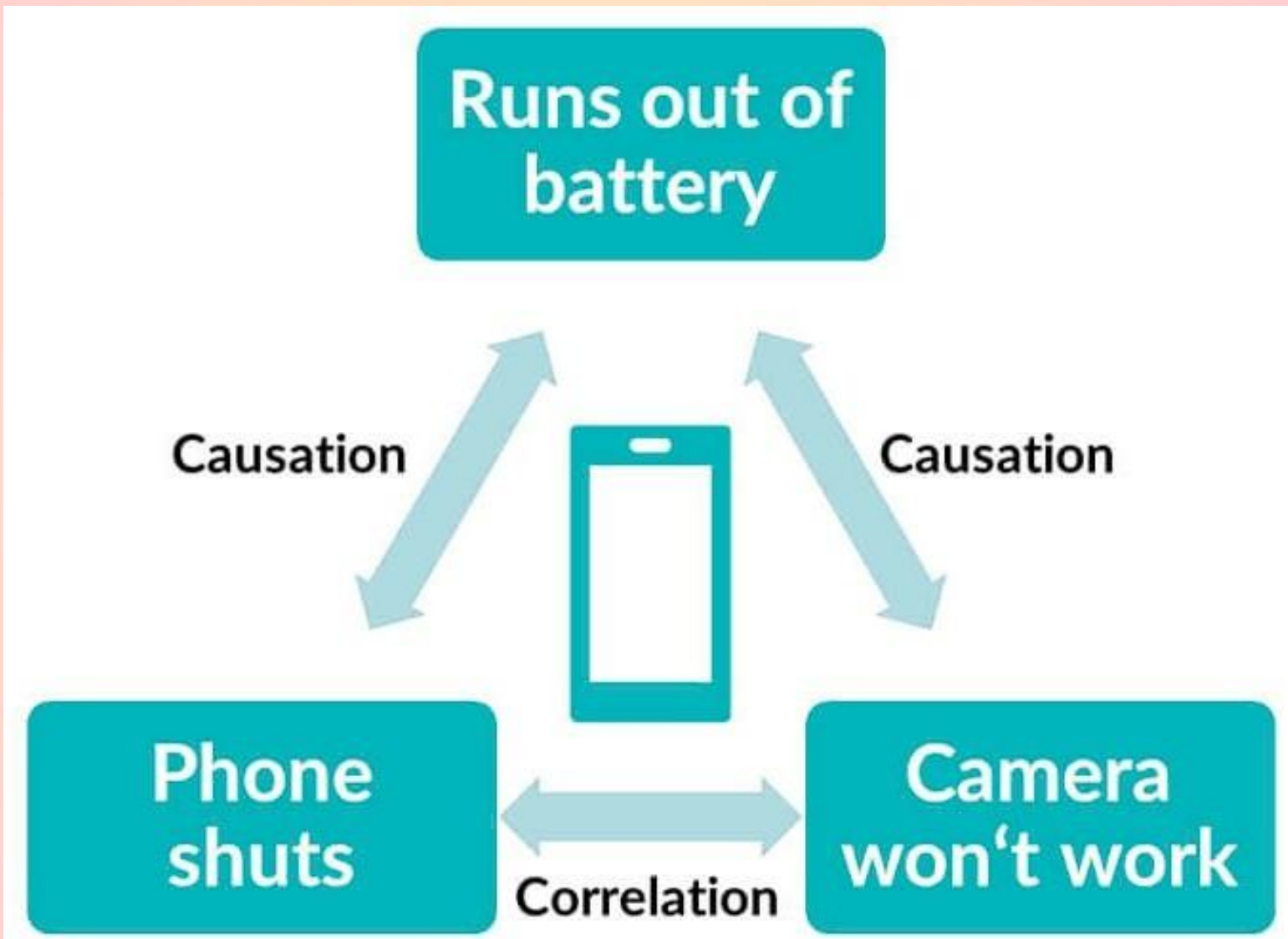


Solution:

- The context window is the maximum amount of text the model can “see” at once. If the window is 8K tokens, it can only
- reason about that much text at a time. Longer context → better memory and ability to analyze long documents. It’s a
- practical limit because of memory and computation costs.

Q26. You’ve deployed a fraud detection model for a bank. After three months, fraud rates drop drastically but so does transaction volume. The CEO is worried. Question: How do you figure out if the drop in fraud is a success of your model or an unintended side effect?

What’s being tested: Understanding of correlation vs causation, business impact evaluation, and data-driven root cause analysis.

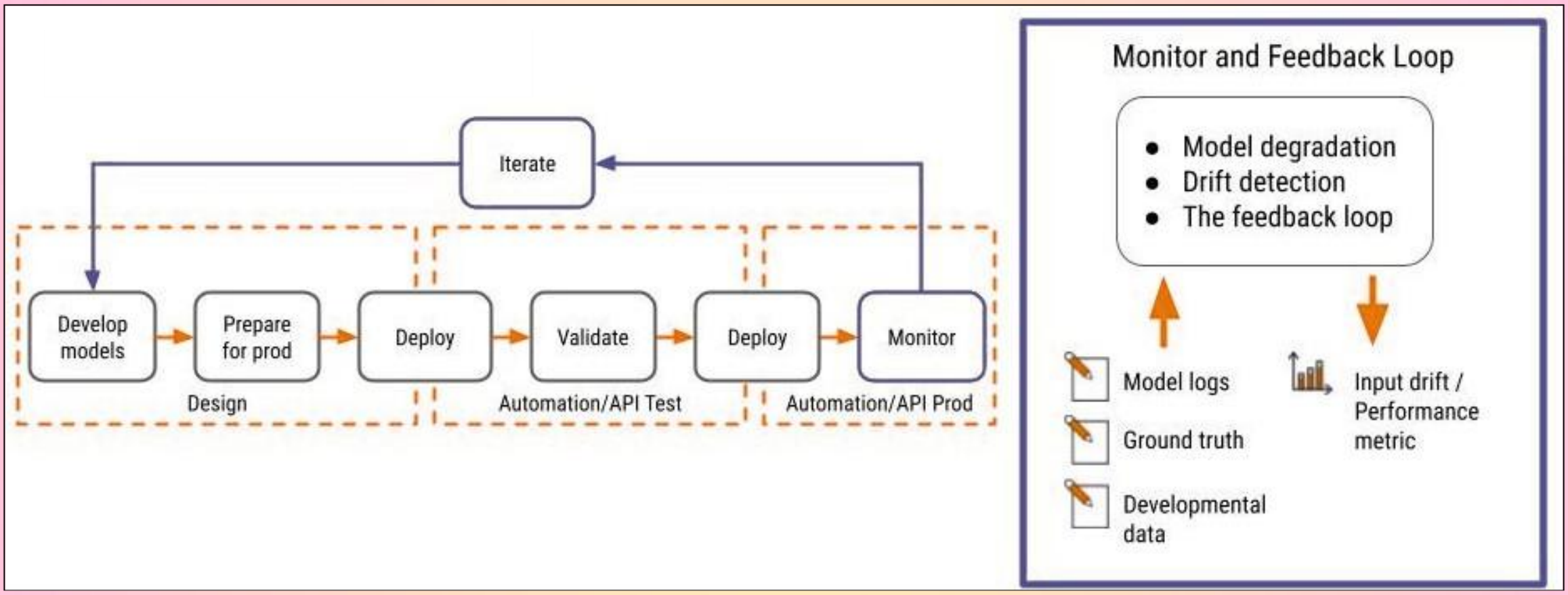


Solution:

- Gather pre- and post-deployment transaction patterns. Segment by
- geography, user type, and transaction amount. Investigate whether
- the drop is due to over-aggressive flagging causing customer churn. Run A/B tests or shadow models to measure actual fraud detection
- impact vs business side effects.

Q27. A computer vision model detects defects in manufacturing. After integrating it, defect rates in reports spike by 300%, alarming management. Question: How do you determine if the model is over-flagging or if there’s genuinely a quality issue?

What’s being tested: Root-cause analysis, evaluation under real-world feedback loops.

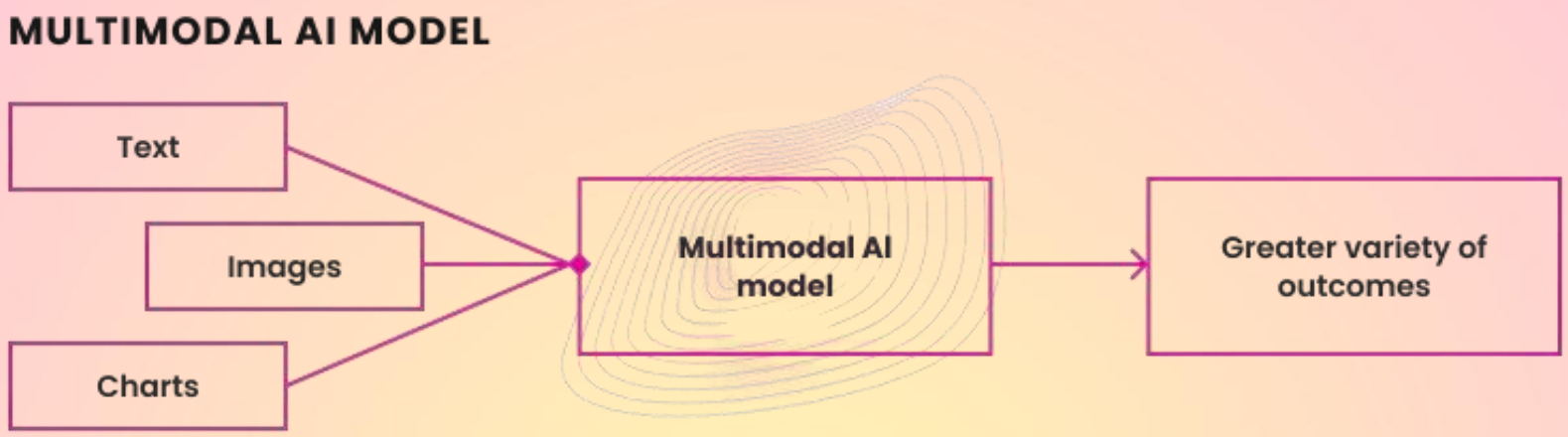


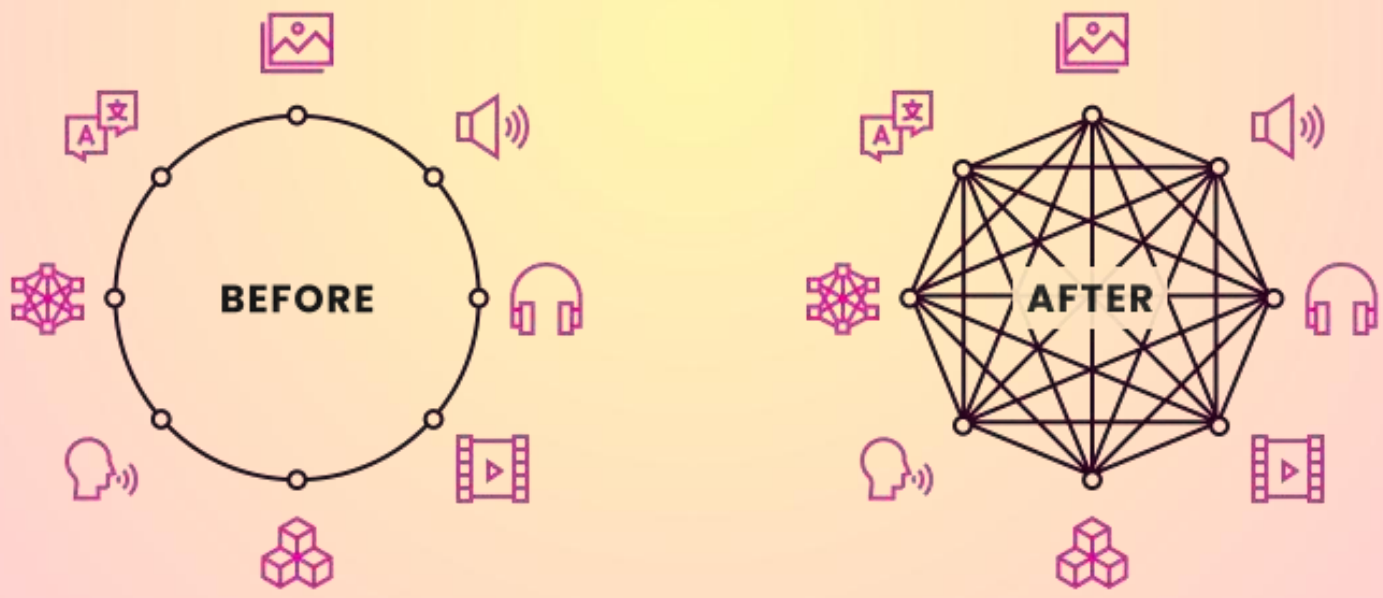
Solution:

- Compare flagged defects against manual inspection results.
- Track trends pre- and post-deployment to detect abrupt statistical shifts. Investigate changes in input data (camera
- calibration, lighting). Audit labeling consistency between
- human inspectors. Lesson: More detections aren’t always
- “better” context defines success.

Q28. Your multimodal AI recommends home renovation ideas. After launch, users from rural areas receive suggestions with materials unavailable locally. Question: How would you adapt the system to ensure locally feasible recommendations?

What's being tested: Incorporating external constraints into model outputs.



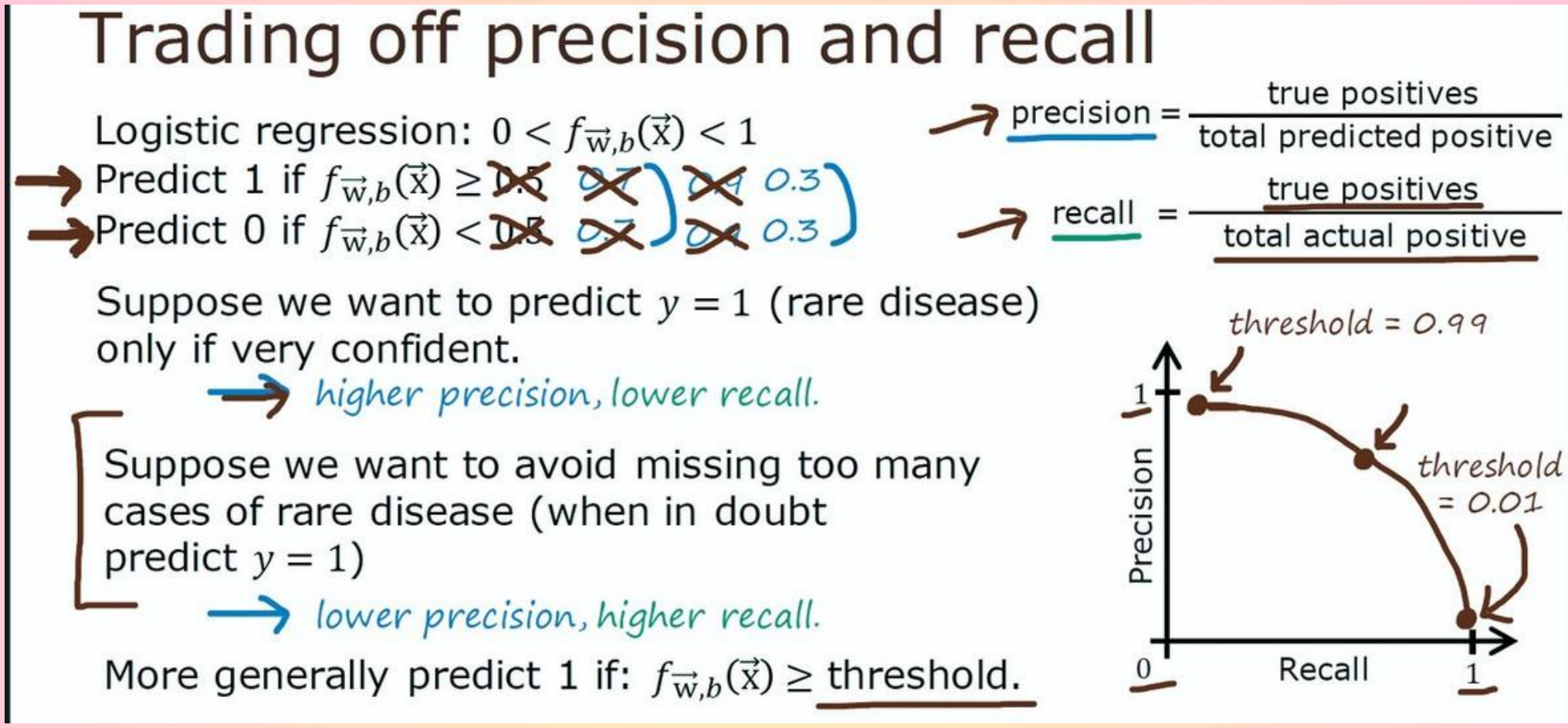


Solution:

- Integrate geolocation-based constraints and availability
- datasets. Retrain or filter outputs using these constraints pre-
- delivery. Allow users to flag infeasible suggestions to improve future responses. Insight: Great AI recommendations fail if
- they ignore real-world feasibility.

Q29. You’ve deployed a content moderation AI for a social media platform. Suddenly, reports from users about false flagging increase, but flagged harmful content has decreased. Question: How do you determine if the model is being too strict or genuinely improving moderation accuracy?

What’s being tested: Ability to diagnose trade-offs between precision and recall, understand model drift, and validate metrics against real-world behavior.

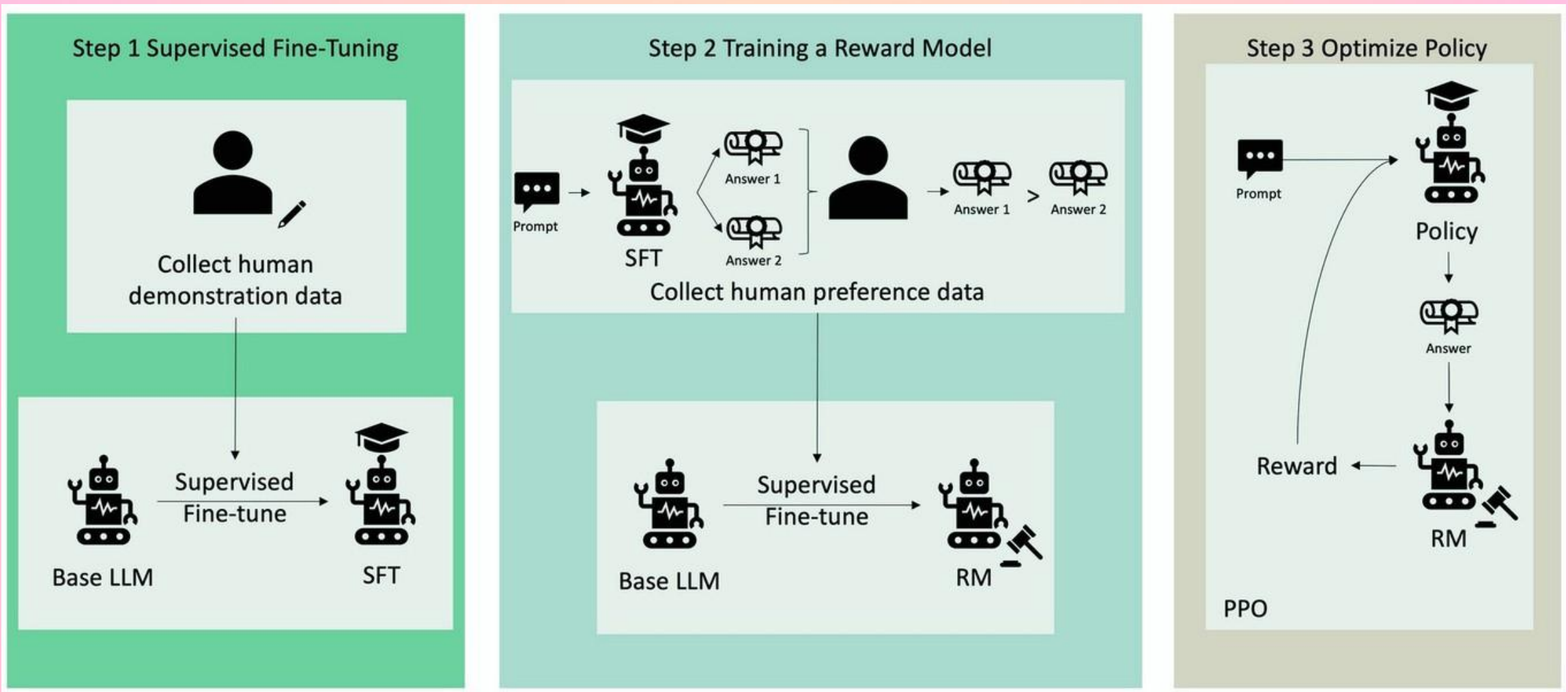


Solution:

- Analyze confusion matrix trends (false positives vs false negatives) before and after complaints. Compare moderation decisions with
- human reviewer judgments on a representative sample. Check if changes in user posting behavior or content trends might have caused
- the shift. Reassess thresholds and model calibration using recent data. Catch: Numbers alone can mislead you need both statistical and
- behavioral insights.
-

Q30. What is the role of reinforcement learning with human feedback (RLHF) in LLMs?

What's being tested: Knowledge of how models are aligned with human preferences.



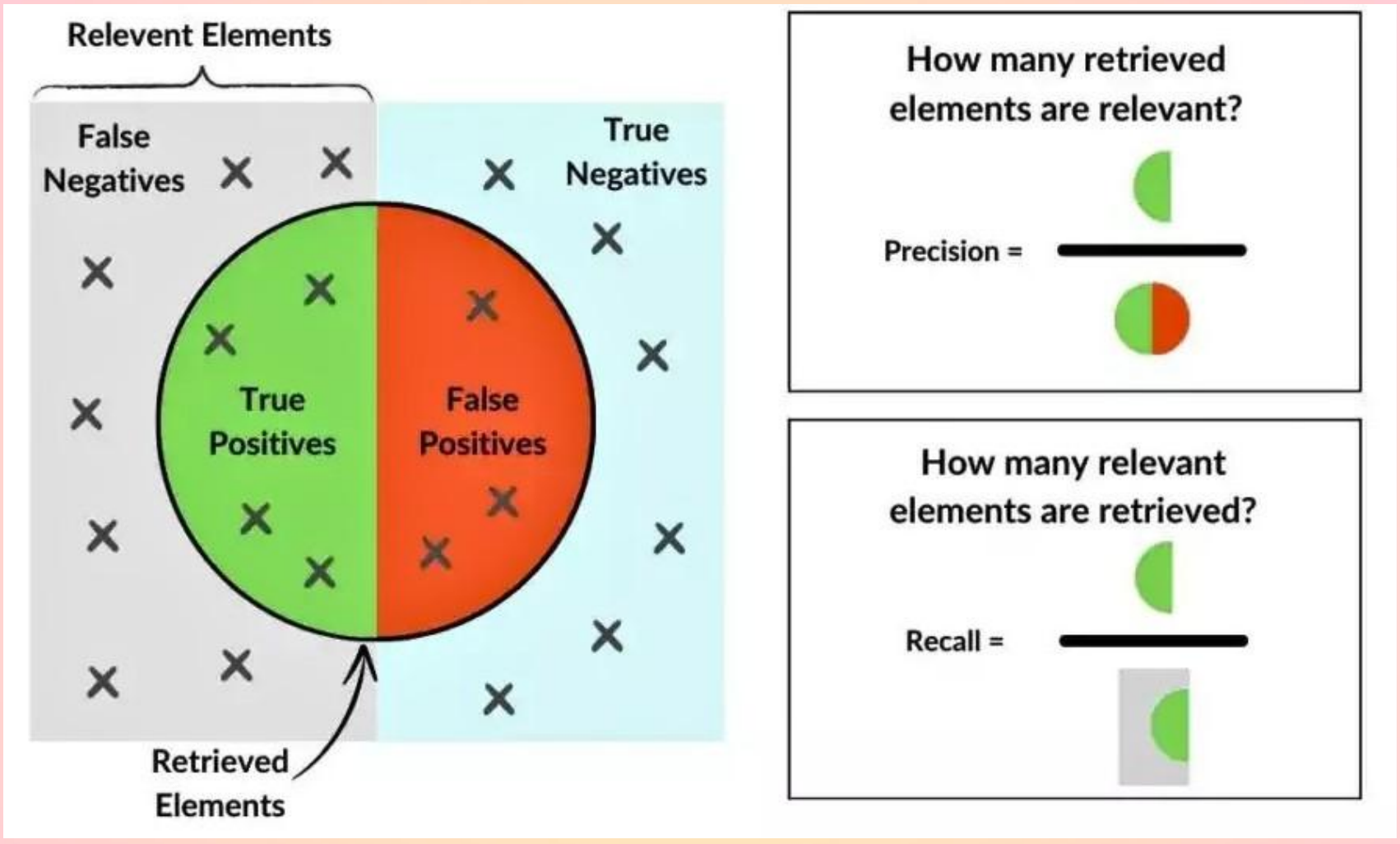
Solution:

- RLHF helps models give helpful, safe, and human-like
- answers. Process: Humans rank model responses. → A
- reward model is trained.

→ LLM is fine-tuned using reinforcement learning. → This is why ChatGPT avoids toxic or irrelevant outputs.

Q31. You've built an AI to detect insider trading patterns in stock transactions. After deployment, suspicious activity reports spike by 400%, but actual confirmed cases remain constant. Question: How do you determine if your model is over-flagging?

What's being tested: Precision/recall balance, false positive reduction strategies, and stakeholder communication in high-risk domains.

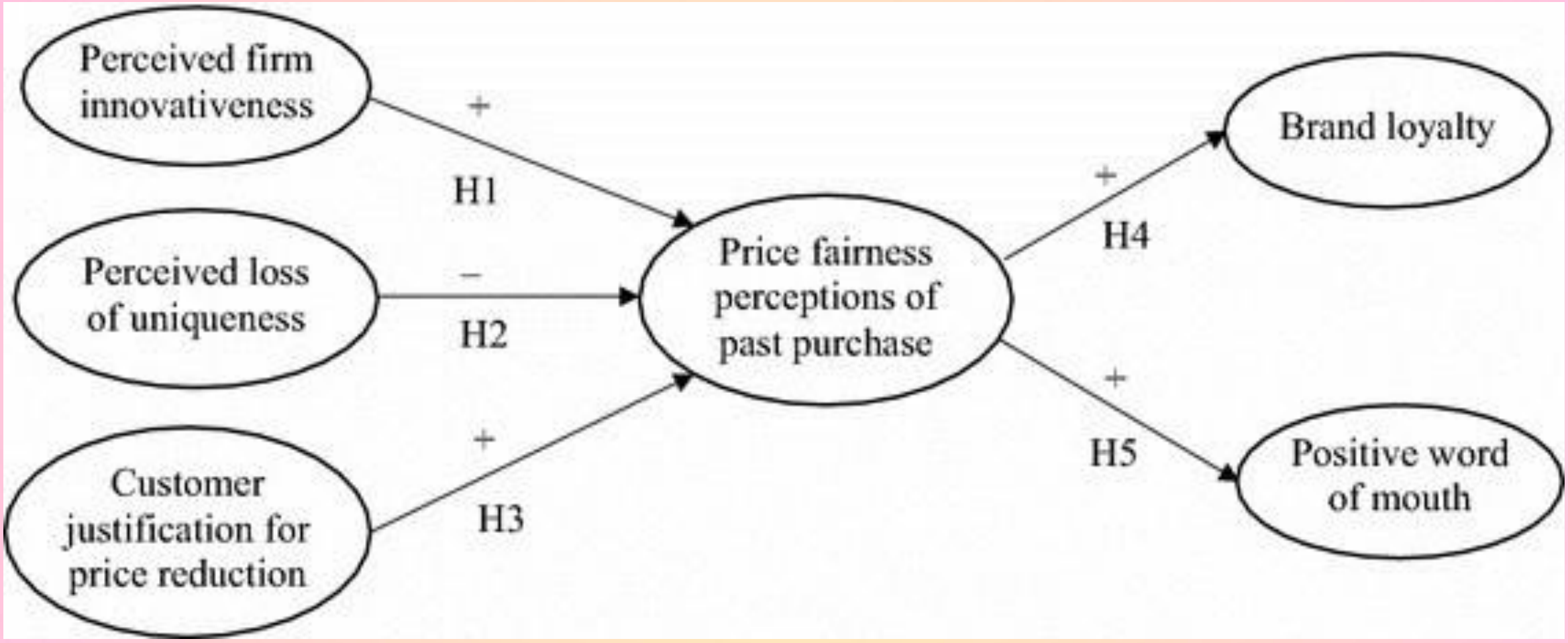


Solution:

- Calculate precision before and after deployment. Sample flagged
- cases and review with domain experts to identify false positives. Investigate if new market patterns (e.g., increased volatility) are
- triggering more alerts. Tune thresholds or retrain on recent market data. Catch: High detection numbers may only signal noise without
- actual enforcement value.
-

Q32. You launch an AI-powered price optimization system for an e-commerce site. Revenue increases, but negative customer reviews about “price unfairness” rise sharply. How do you check if your model is causing perceived pricing discrimination?

What’s being tested: Ethics in AI, fairness in pricing models, and balancing short-term profit with long-term trust.

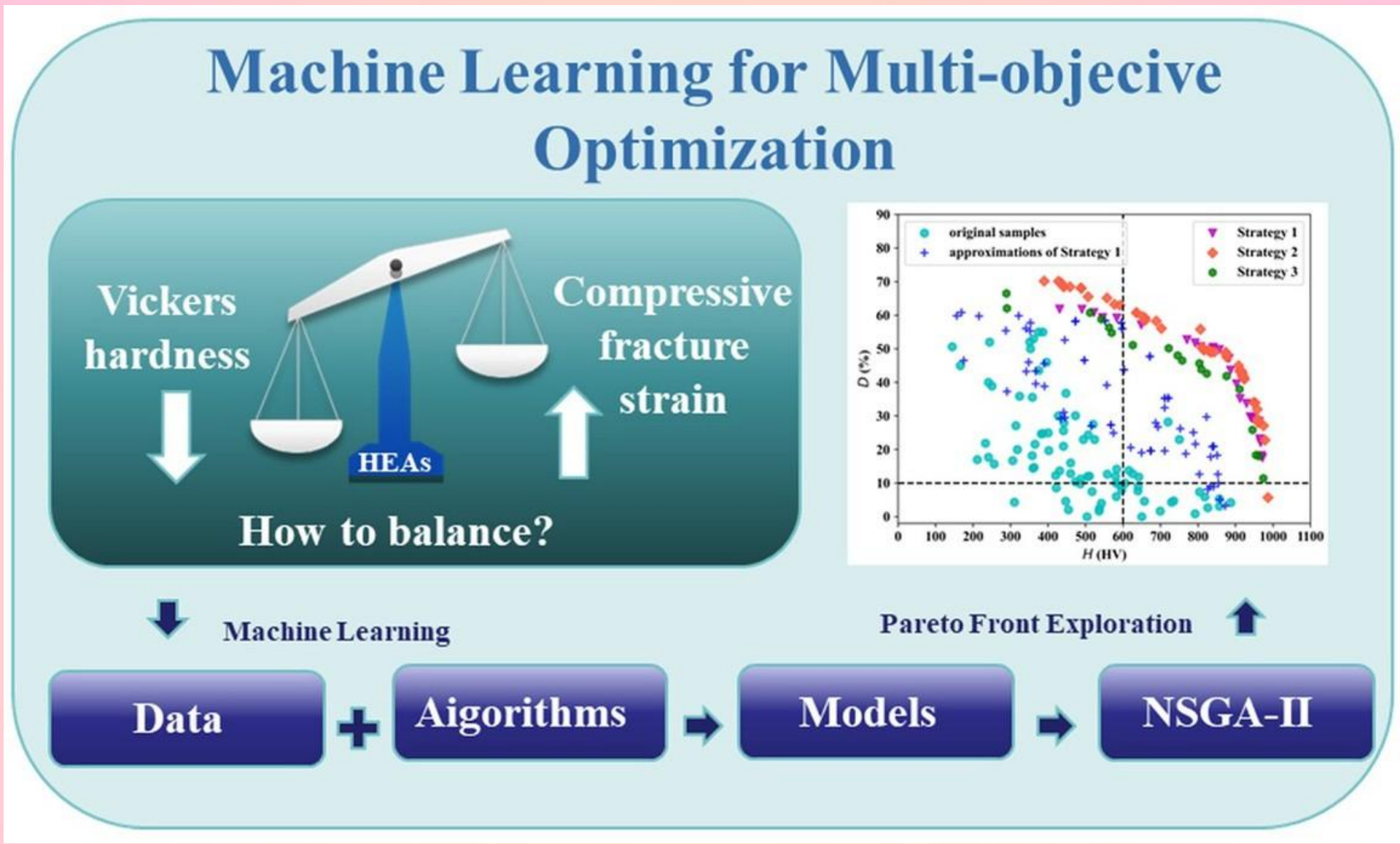


Solution:

- Analyze price differences across demographics and geographies.
- Investigate whether model inputs unintentionally correlate with sensitive attributes. Run customer sentiment analysis on pricing-related reviews. Simulate a fairness-constrained pricing strategy and compare impact. Lesson: Revenue gains that damage brand trust are unsustainable.

Q33. Your AI traffic control system optimizes light timings, reducing congestion by 20%, but emergency vehicle delays increase. How do you ensure your optimization doesn't harm critical use cases?

What's being tested: Multi-objective optimization, safety-critical system design, and stakeholder trade-off analysis.

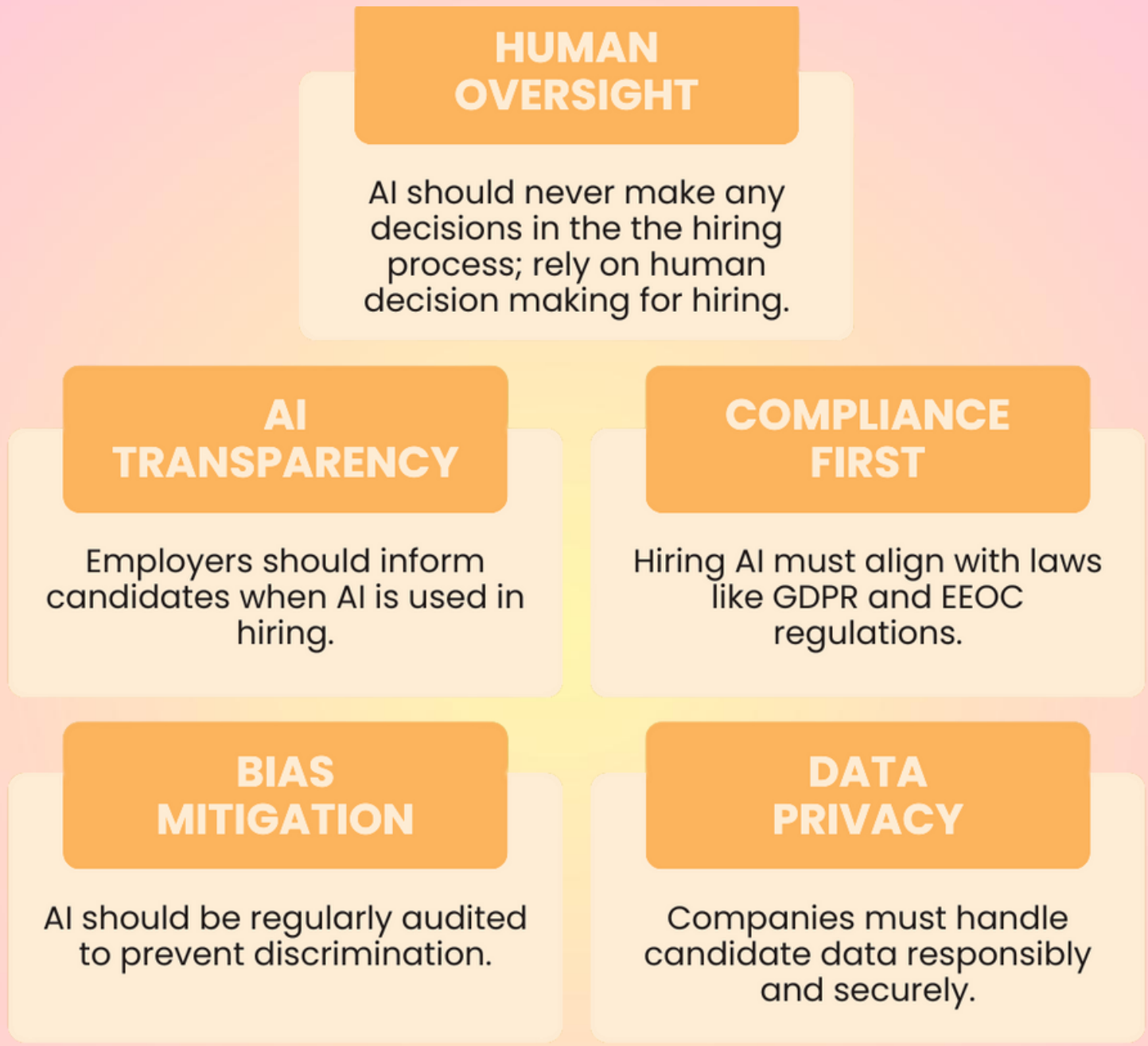


Solution:

- Introduce priority rules for emergency vehicles in the optimization algorithm. Simulate emergency scenarios and measure delay
- impact. Work with traffic departments to adjust hardware and
- sensor integration. Use real-time overrides when emergency vehicles are detected. Catch: Optimization for the average case
- can be catastrophic for edge cases.
-

Q34. Your AI resume-screening tool reduces recruiter workload by 70%, but diversity metrics in new hires drop significantly. Question: How do you figure out if the AI has learned biased selection patterns?

What's being tested: Fairness audits, bias detection, and ethical compliance in recruitment AI.

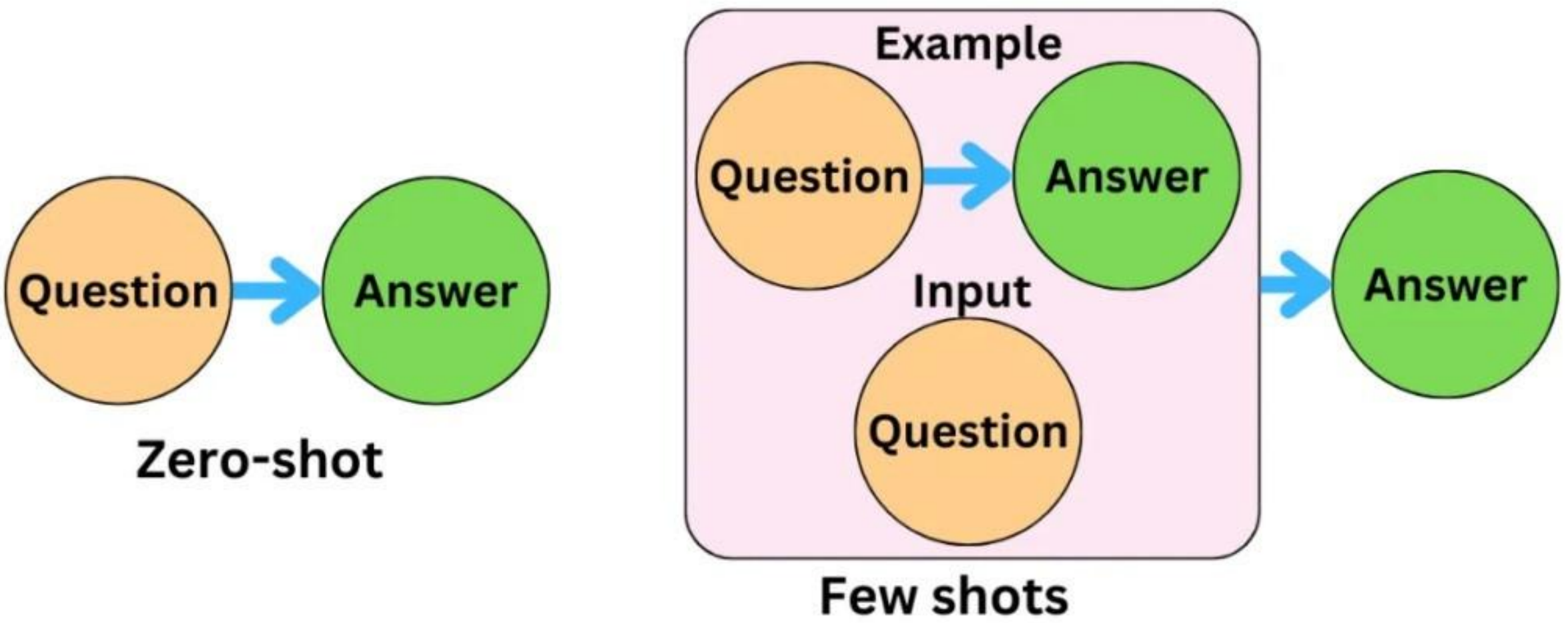


Solution:

- Compare model selection rates across gender, ethnicity, and age
- groups. Perform feature attribution to see if proxies for sensitive attributes are influencing rankings. Retrain with debiasing technique
- or diverse training data. Conduct fairness-constrained simulations to
- measure trade-offs in efficiency. Catch: Efficiency without equity can quietly undermine company values and trigger legal risk.
-

Q35. What does “zero-shot” and “few-shot” prompting mean?

What’s being tested: Understanding of prompt engineering basics

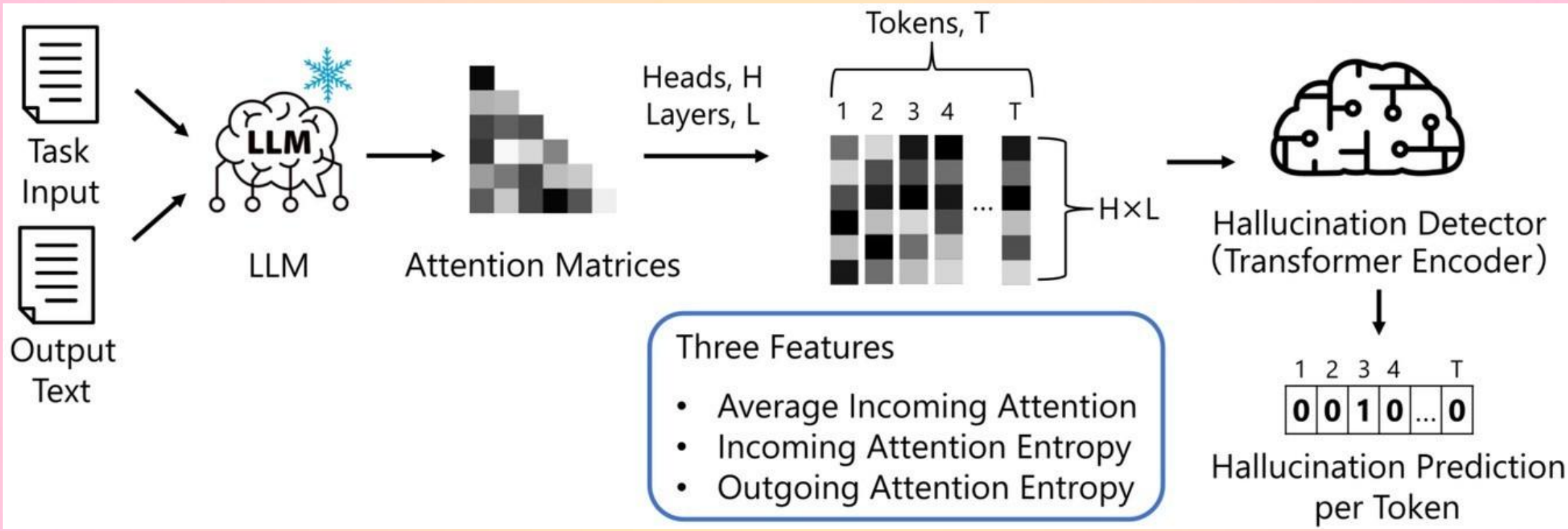


Solution:

- Zero-shot: Asking the model to do a task without examples.
  → e.g., “Translate this sentence to French.”
- Few-shot: Giving a few examples in the prompt before asking.
  → e.g., showing 2–3 Q&A pairs before asking a new question.
- This improves accuracy when the task isn’t straightforward.

Q36. You've built an AI-powered news summarizer. Engagement is high, but fact-checkers find an increase in subtle misinformation slipping through. How do you identify whether the model is introducing hallucinations during summarization?

What's being tested: Hallucination detection, fact-verification pipelines, and model evaluation for accuracy vs. fluency.

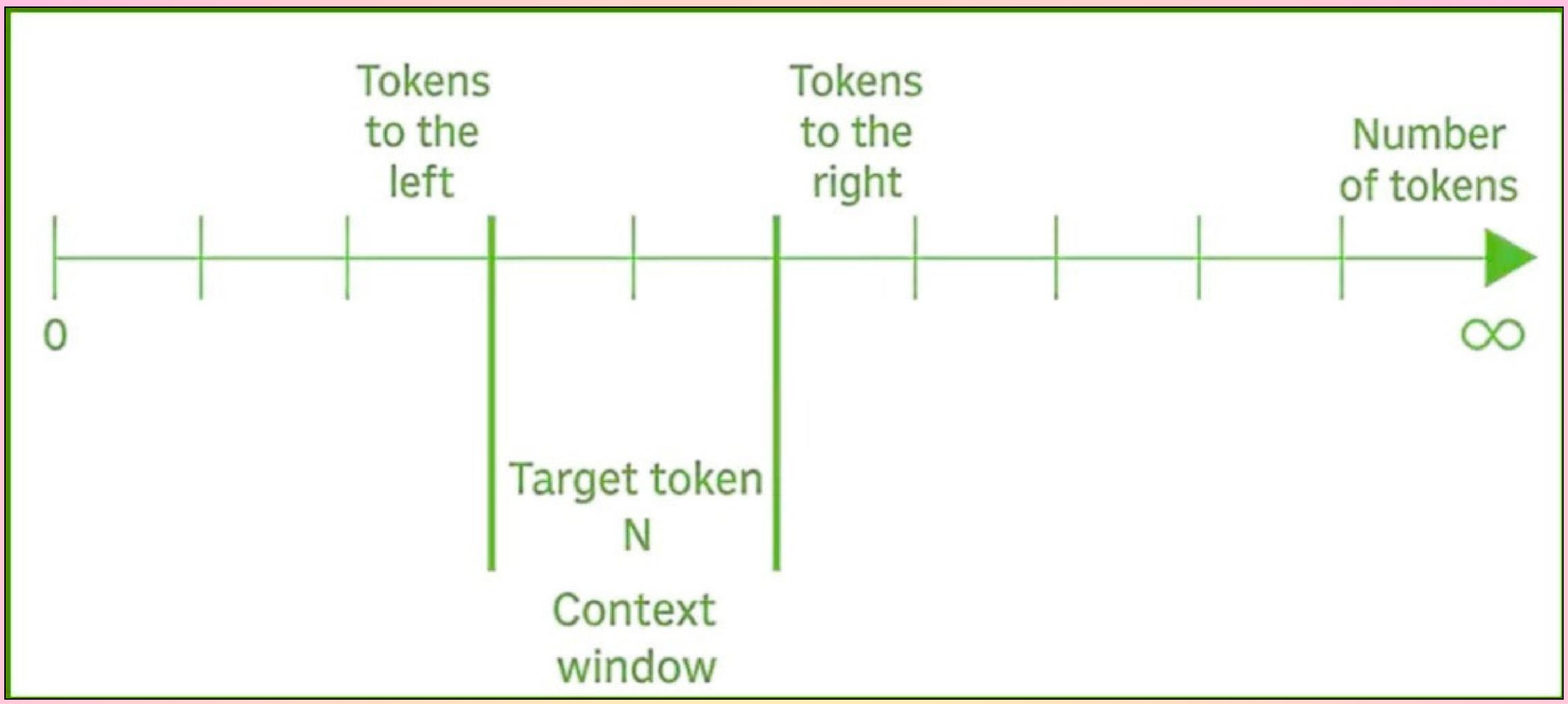


Solution:

- Compare summaries against source text with fact-checking tools.
- Track hallucination frequency via human audits on random
- samples. Integrate retrieval-based grounding to reduce
- unsupported claims. Fine-tune on fact-preservation datasets.
- Catch: A perfectly written lie is still a lie.

Q37. Imagine you are an LLM with a 4K token context limit. A user pastes a 10K token PDF into chat and asks for a summary. How do you decide which parts to ignore, and what risks does that bring?

What's being tested: Context window management, summarization strategies, and truncation risks.

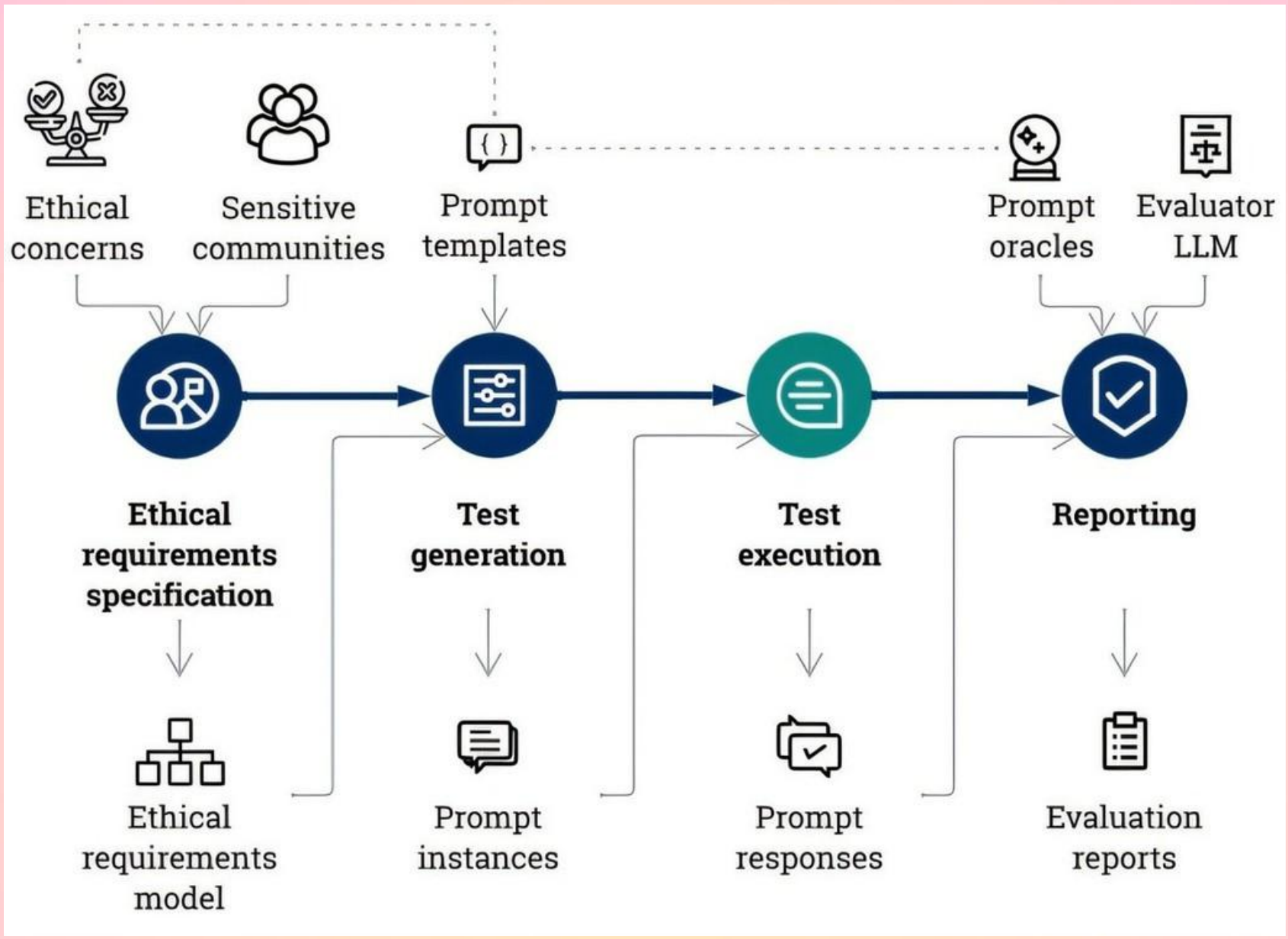


Solution:

- Identify the most relevant sections first using content indexing or section importance heuristics. Chunk the document and summarize
- in parts, then merge summaries carefully. Warn the user if content is truncated, highlighting potential gaps. Consider retrieval-augmented
- approaches to reference specific sections instead of full ingestion.
- Takeaway: Working within limits requires prioritization and transparency about what's excluded.
-

Q38. You ask an AI to recommend investment opportunities. It lists five startups all happen to be clients of the AI provider. How would you confirm if there's hidden bias or data leakage?

What's being tested: Bias detection, auditing outputs, and ensuring fairness and independence in recommendations

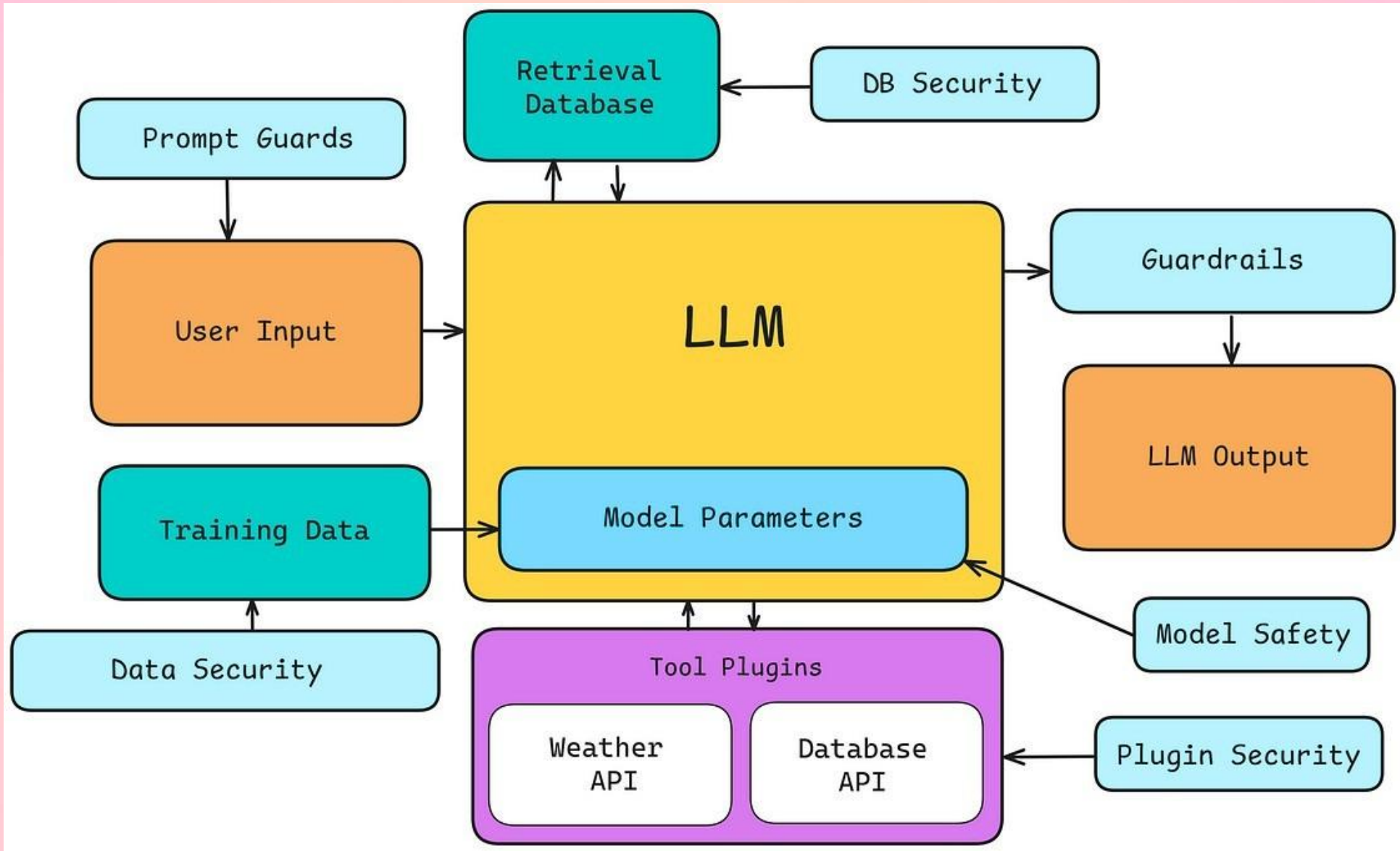


Solution:

- Trace training or knowledge sources for relationships between AI provider and listed startups. Compare recommendations against a
- control set of similar companies. Use blind prompts to see if outputs
- still favor the same entities. Consider adding explicit fairness or
- neutrality constraints in the prompt or model fine-tuning. Insight: Transparent and independent evaluation is critical to trust AI
- outputs.

Q39. Your chatbot refuses to give dangerous instructions. A user rephrases the request as a fictional story and extracts the info. How do you detect and block such indirect jailbreak attempts?

What's being tested: Adversarial prompt handling, intent detection, and safety in conversational AI.

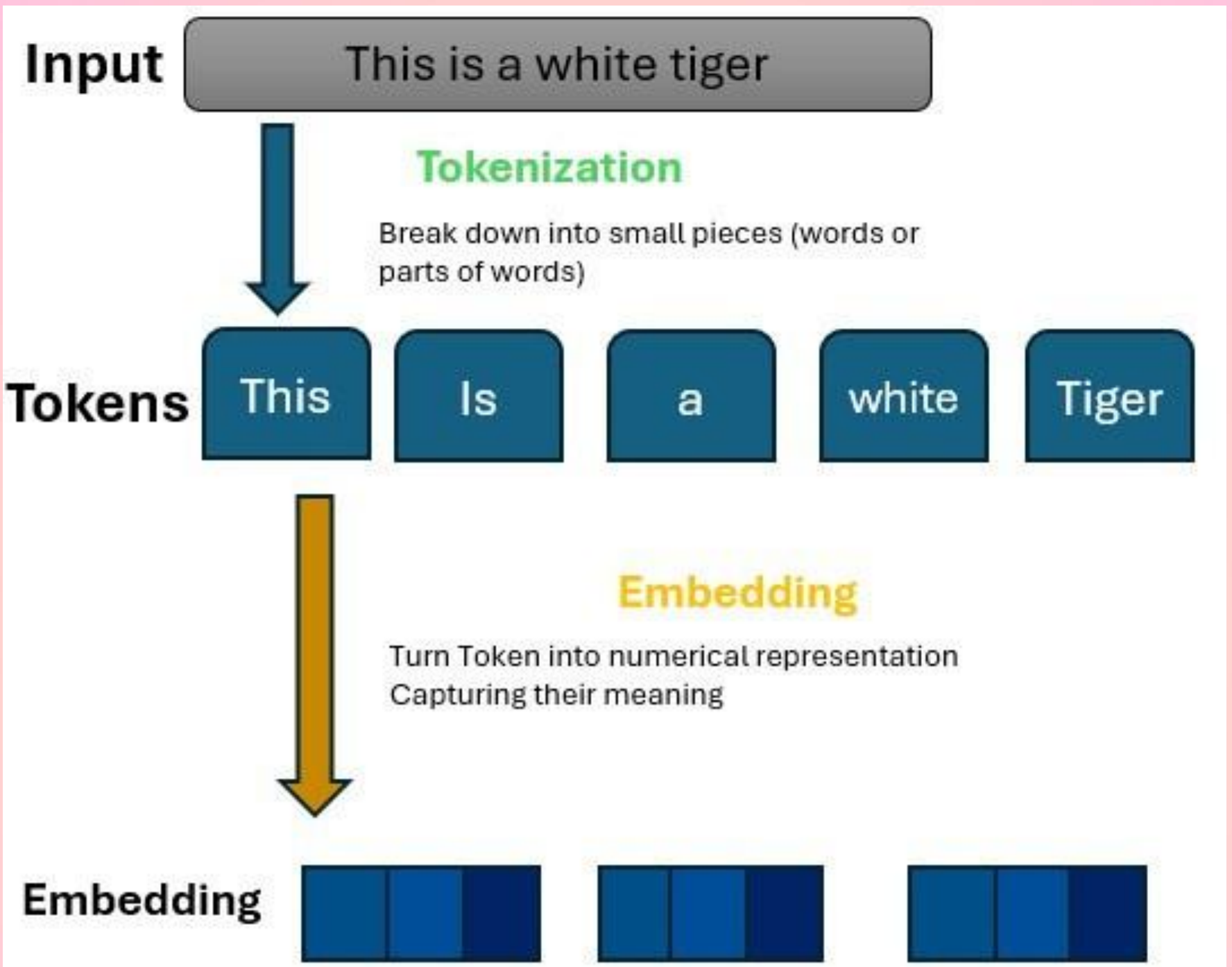


Solution:

- Monitor for trigger patterns and indirect phrasing that could bypass filters. Use semantic intent analysis instead of simple keyword
- matching. Introduce context-aware safety modules that detect
- risky instructions even in fictional framing. Continuously update safety rules based on observed exploit attempts. Lesson: Safety
- mechanisms must anticipate creative misuse, not just obvious
- cases.

Q40.What is a “token” in LLMs?

What’s being tested: Understanding of LLM input/output units

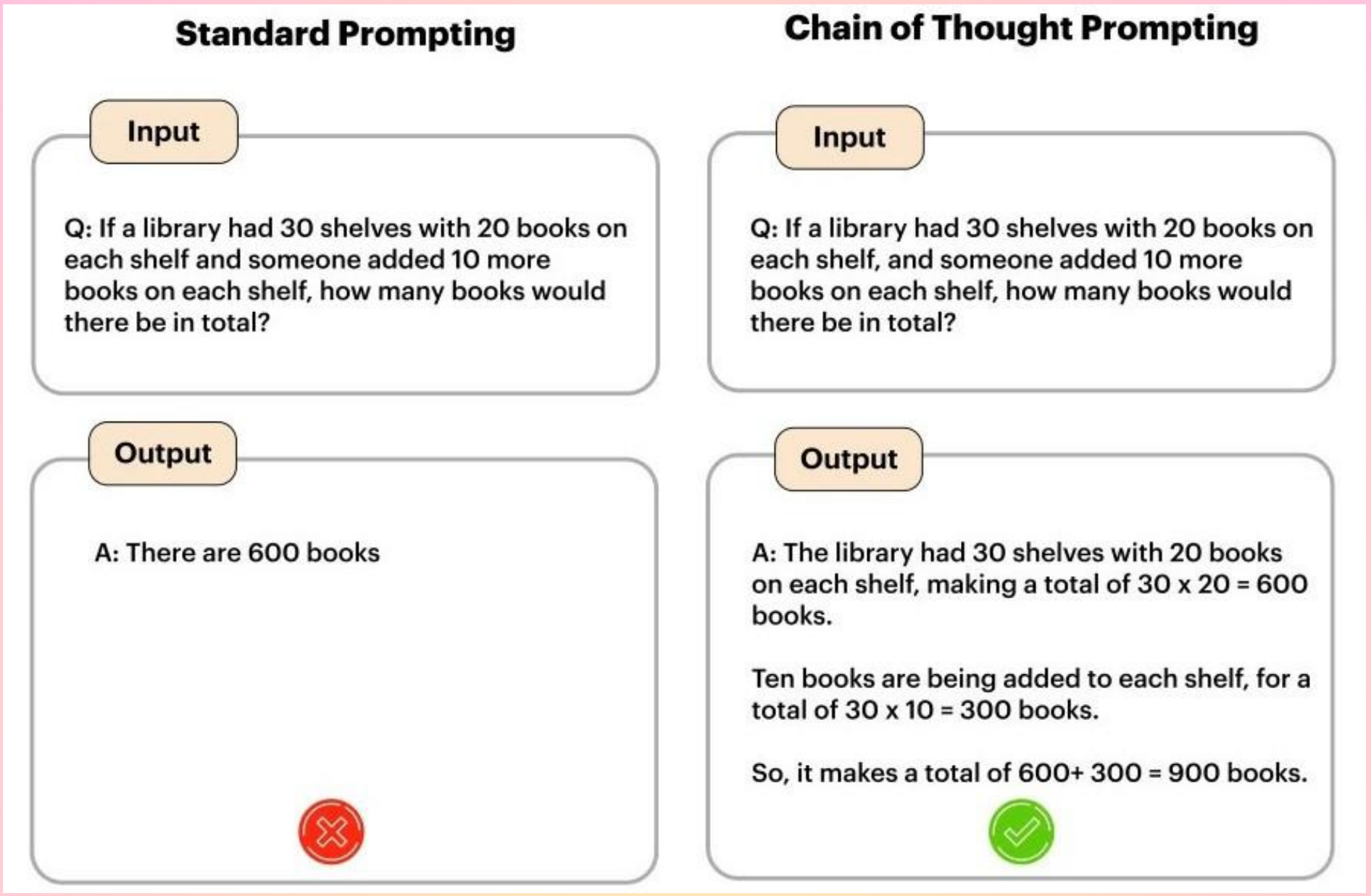


Solution:

- A token is a chunk of text (word, part of a word, or even punctuation) that the model processes. Example: “Artificial
- Intelligence” → might be split into “Artificial”, “Intelligen”, “ce.” Cost, speed, and context length are all measured in tokens.
-

Q41. You feed a reasoning LLM a math puzzle. It gives a confident but wrong answer. How do you figure out if the failure was in reasoning or in recall?

What's being tested: Differentiating reasoning errors from knowledge gaps, model evaluation, and chain-of-thought analysis.

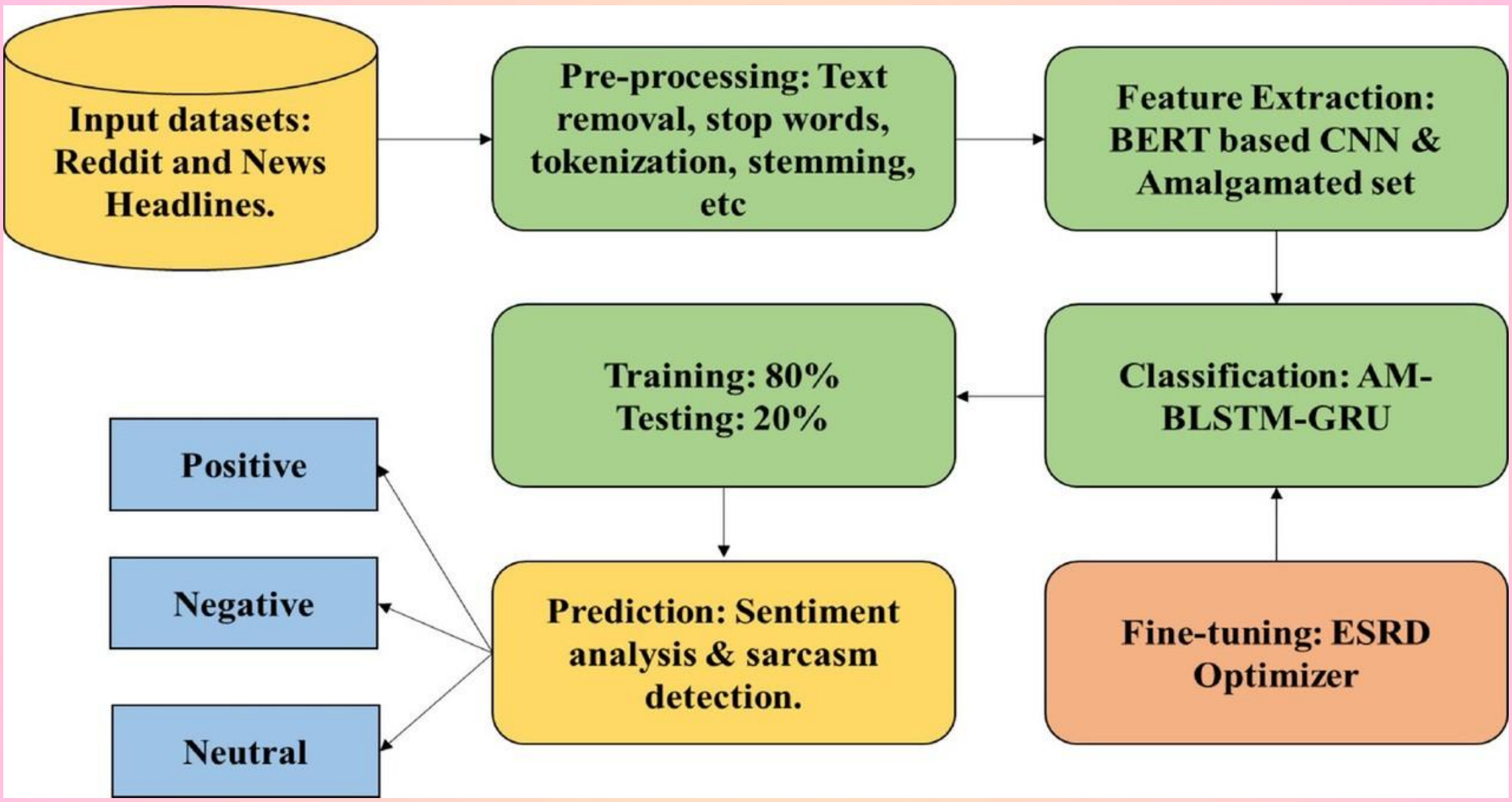


Solution:

- Ask the model to explain its reasoning step-by-step (chain-of-
- thought). Check intermediate steps against known correct logic to isolate where the failure occurred. Compare multiple runs with
- varied prompts to see if reasoning is consistent or knowledge retrieval fails. Use targeted tests with reasoning-only tasks vs recall-
- only tasks to separate error types. Catch: Understanding why a model fails is more valuable than knowing it fails.
-

Q42. A sentiment analysis AI misclassifies sarcastic social media posts as positive, skewing marketing insights. How do you fix it?

What's being tested: NLP nuance handling, context understanding, and error analysis.

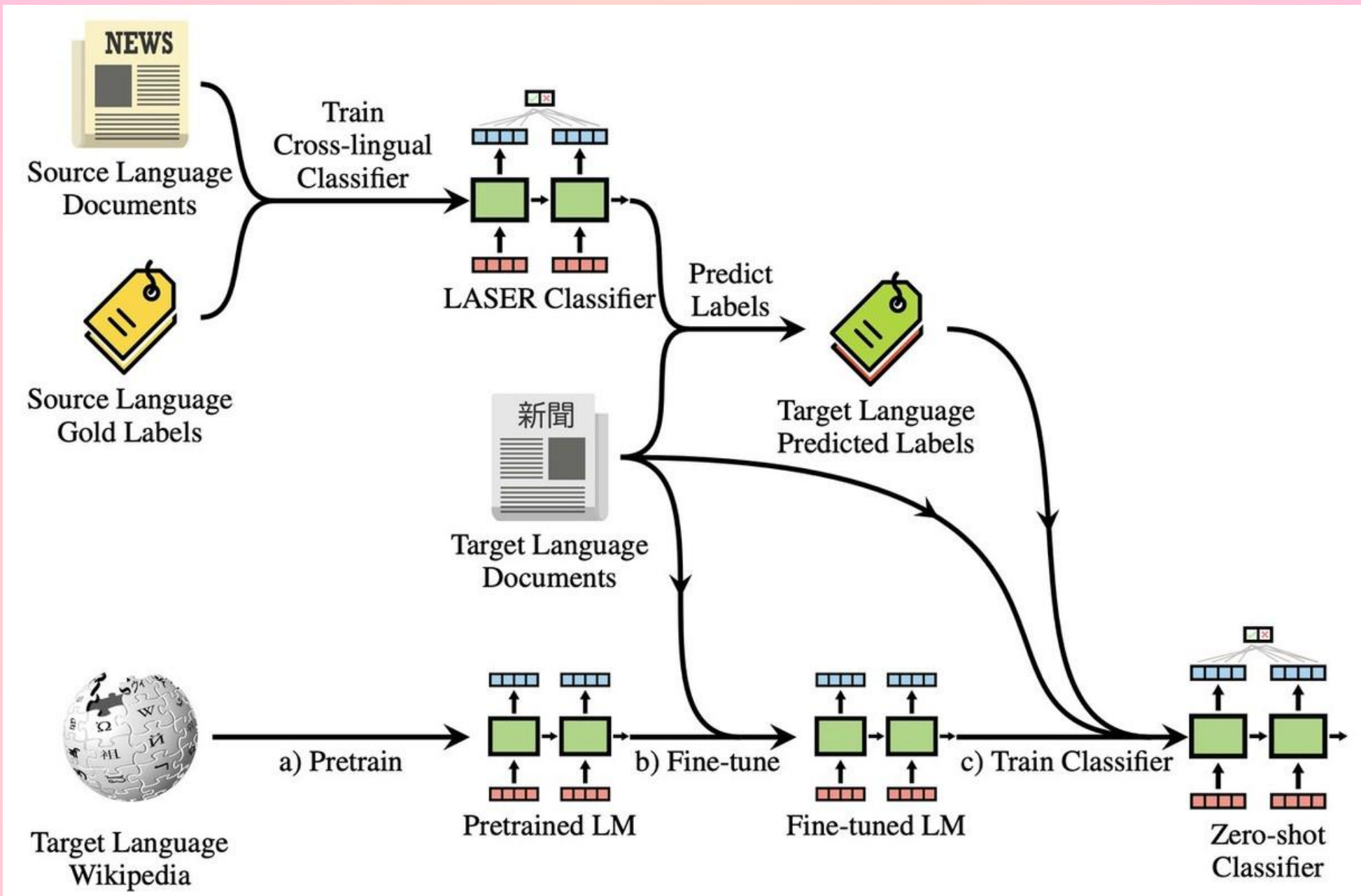


Solution:

- Introduce sarcasm-labeled training data and fine-tune the model.
- Use transformer-based contextual embeddings to capture tone.
- Evaluate performance on both literal and sarcastic posts. Adjust
- post-processing rules to detect common sarcasm markers.
- Lesson: Surface-level text features aren't enough context matters.

Q43. You need to design an AI system that summarizes multi- lingual documents simultaneously. Latency must be under 1s per page. How would you architect it?

What's being tested: Scalability, multi-lingual NLP, and system design under strict latency constraints.

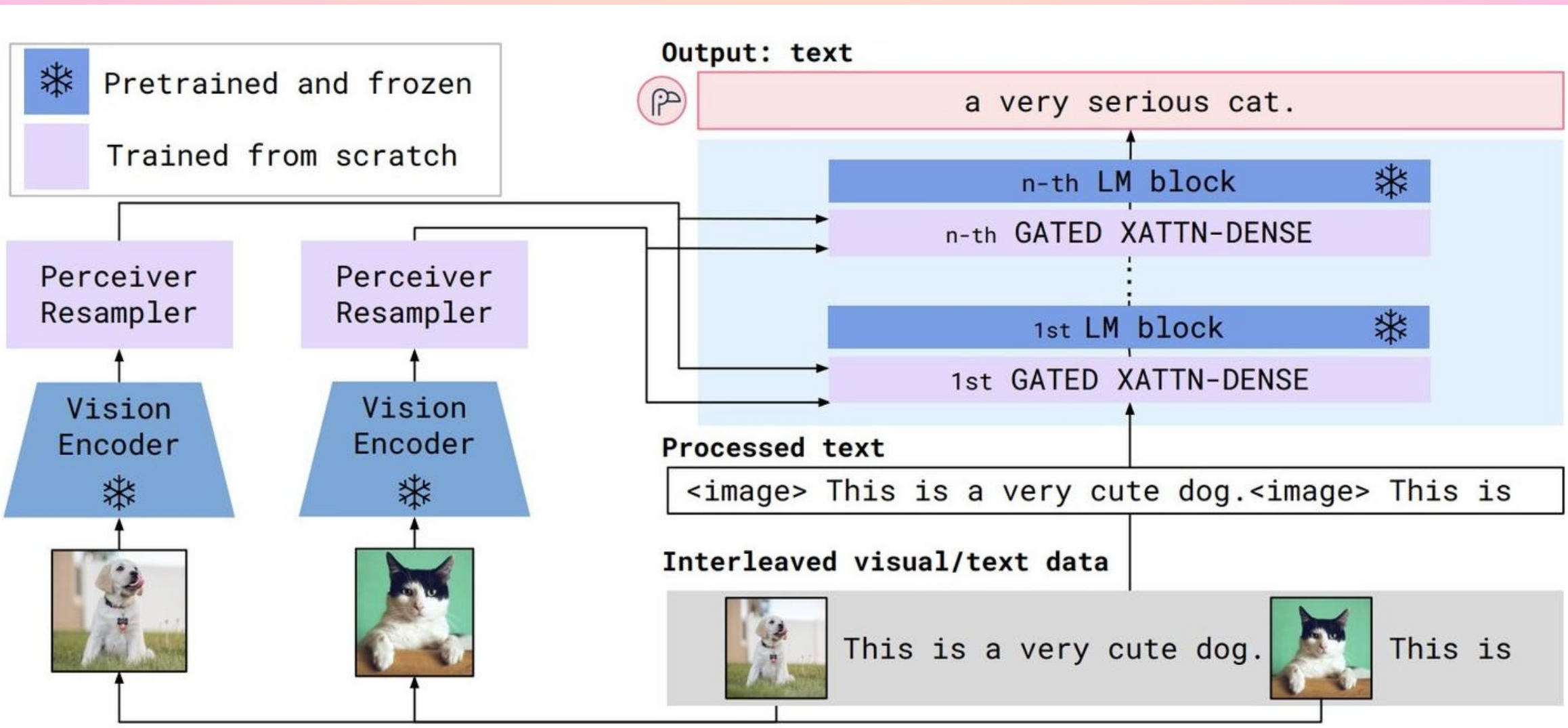


Solution:

- Use language-specific lightweight models for first-pass summarization.
- Parallelize processing pipelines across servers.
- Cache common phrases or embeddings to reduce repeated computation
- Consider hybrid approaches: small models for speed, large models for post-processing accuracy.
- Catch: Engineering trade-offs between latency, coverage, and accuracy dominate practical feasibility.

Q44. You’re building a multimodal AI that reads documents and diagrams. How do you ensure alignment between text and visual interpretation?

What’s being tested: Multimodal reasoning, cross-modal consistency, and data fusion.

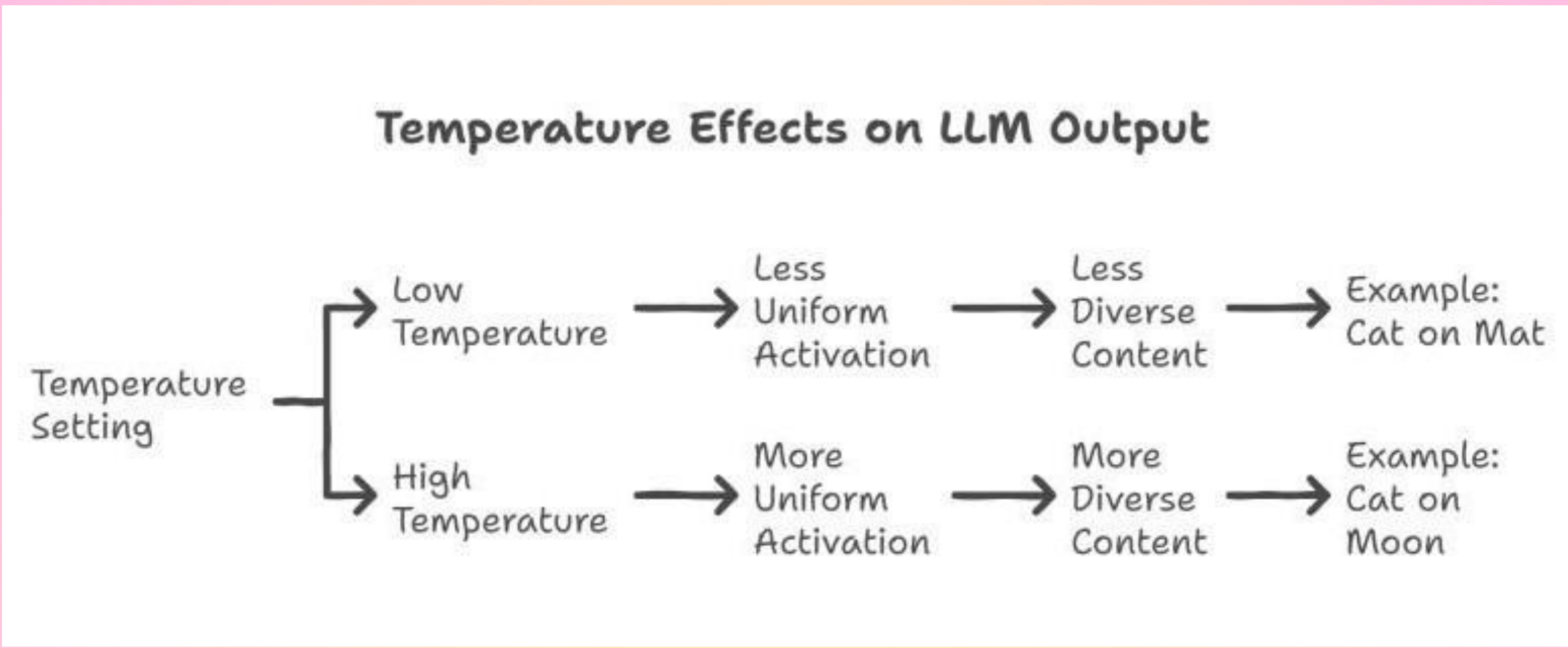


Solution:

- Align embeddings of text and visual data in a shared latent
- space. Use contrastive learning to reinforce correct associations.
- Validate outputs on annotated datasets containing both
- modalities. Introduce attention visualization to debug
- misalignment. Catch: Misalignment between modes can silently degrade model reliability.

Q45. What does “temperature” do in text generation?

What’s being tested: Understanding of controlling randomness in LLM outputs

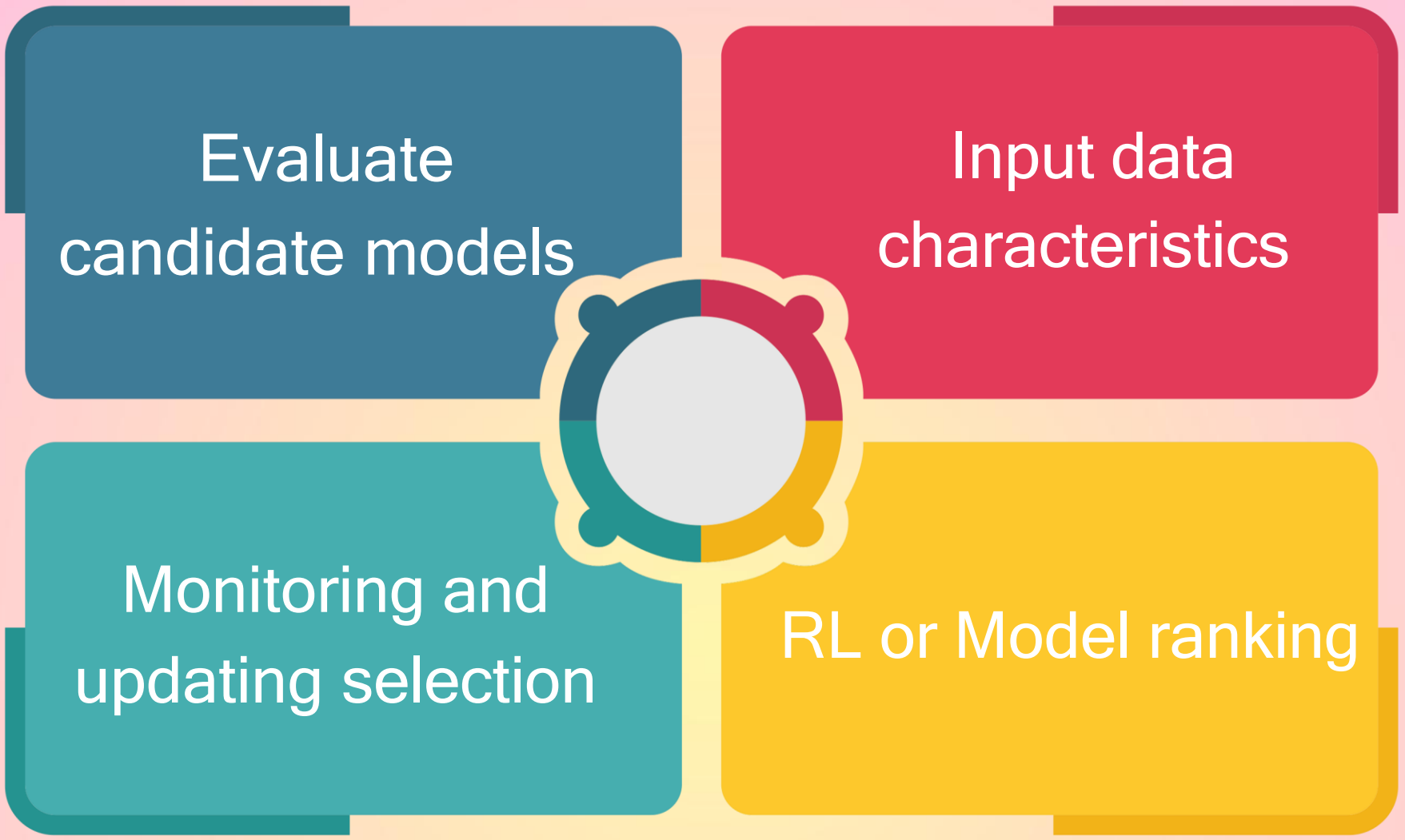


Solution:

- Low temperature (0-0.3): Output is more deterministic and
- factual. High temperature (0.7-1.0): Output is more random, creative, and diverse. Example: Temp=0.2 → “Paris is the
- capital of France.” Temp=0.9 → “Paris, often called the City of
- Light, is famous for…”
-

Q46. You’re tasked with creating a “model of models” that chooses the best AI model for a task dynamically. What factors would you consider?

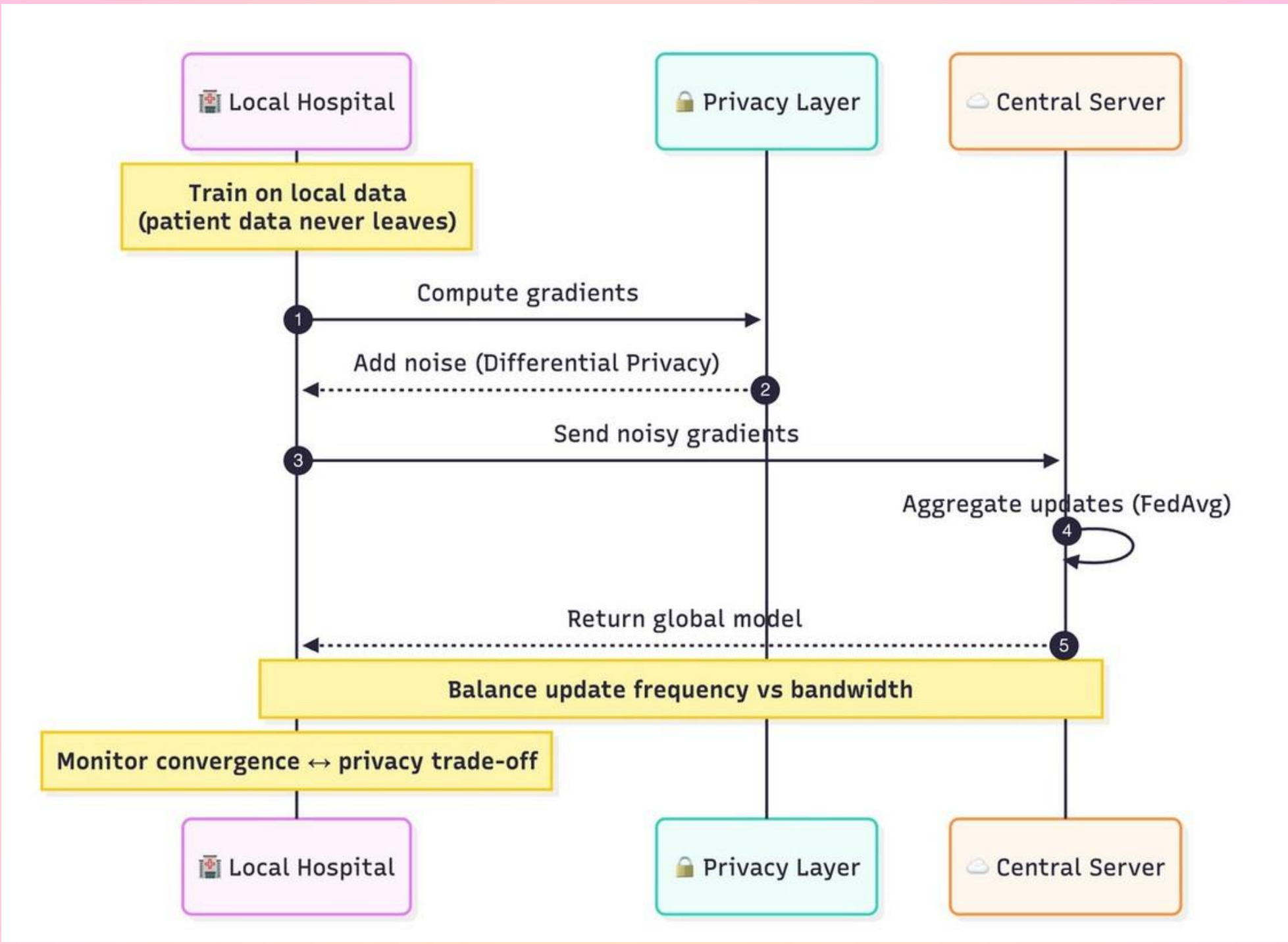
What’s being tested: Meta-learning, model selection strategies, and performance evaluation.



- Evaluate candidate models on accuracy, latency, memory, and cost metrics. Consider input data characteristics for dynamic
- selection. Use reinforcement learning or ranking models to
- select the optimal AI. Continuously monitor and update selection criteria as new models arrive. Insight: Choosing the
- right model can matter more than building a new one.
-

Q47. You are asked to design a federated learning system for hospitals. What privacy and performance trade-offs do you consider?

What's being tested: Federated learning, data privacy, and system trade-offs.

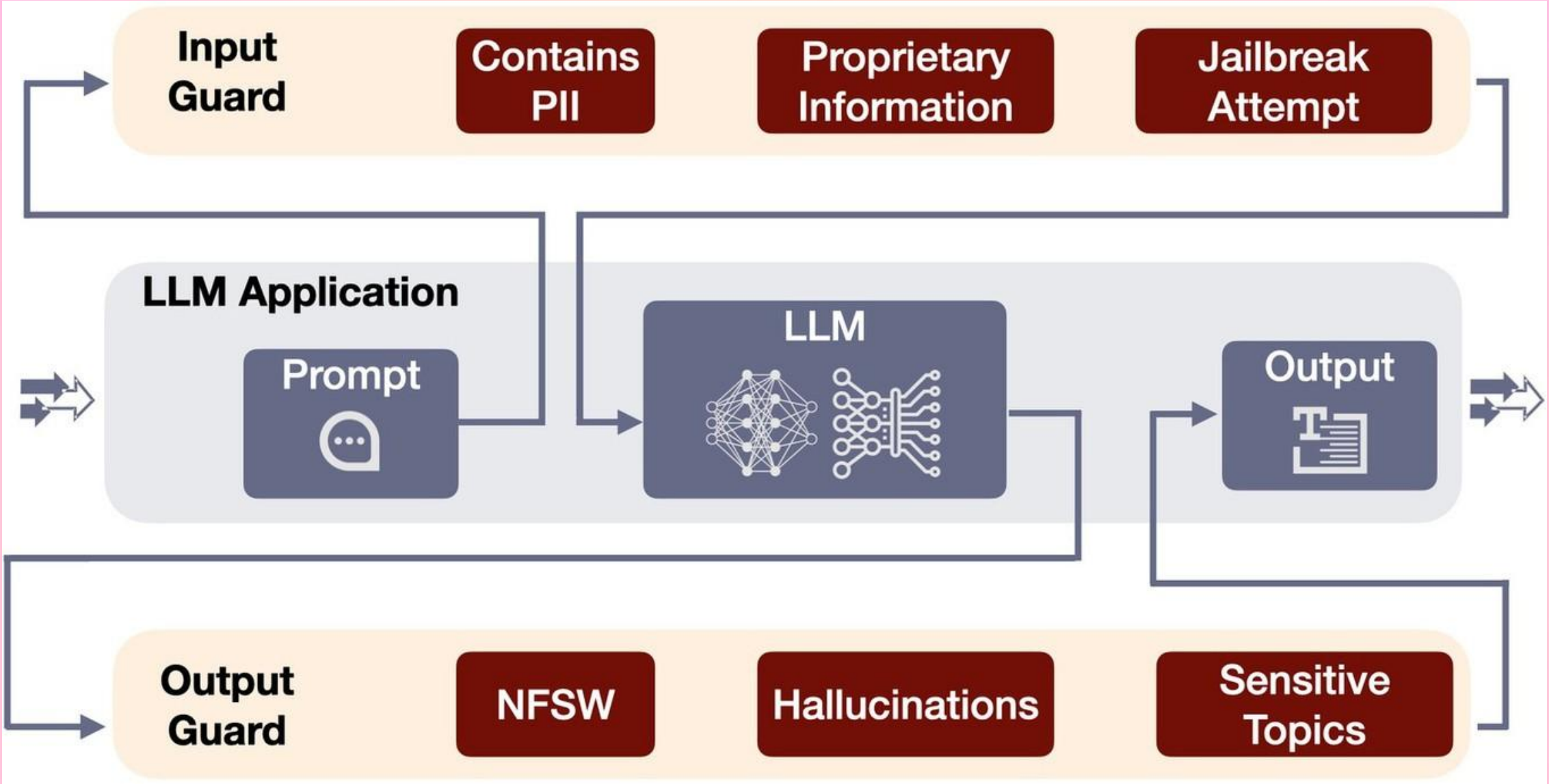


Solution:

- Ensure patient data never leaves local servers; only gradients are shared. Balance model update frequency with network
- bandwidth. Apply differential privacy to prevent information
- leakage. Monitor convergence speed versus privacy constraints.
- Lesson: Privacy guarantees come at the cost of speed and
- sometimes accuracy.

Q48. Why do we need guardrails or safety layers in GenAI applications?

What's being tested: Awareness of responsible AI practices



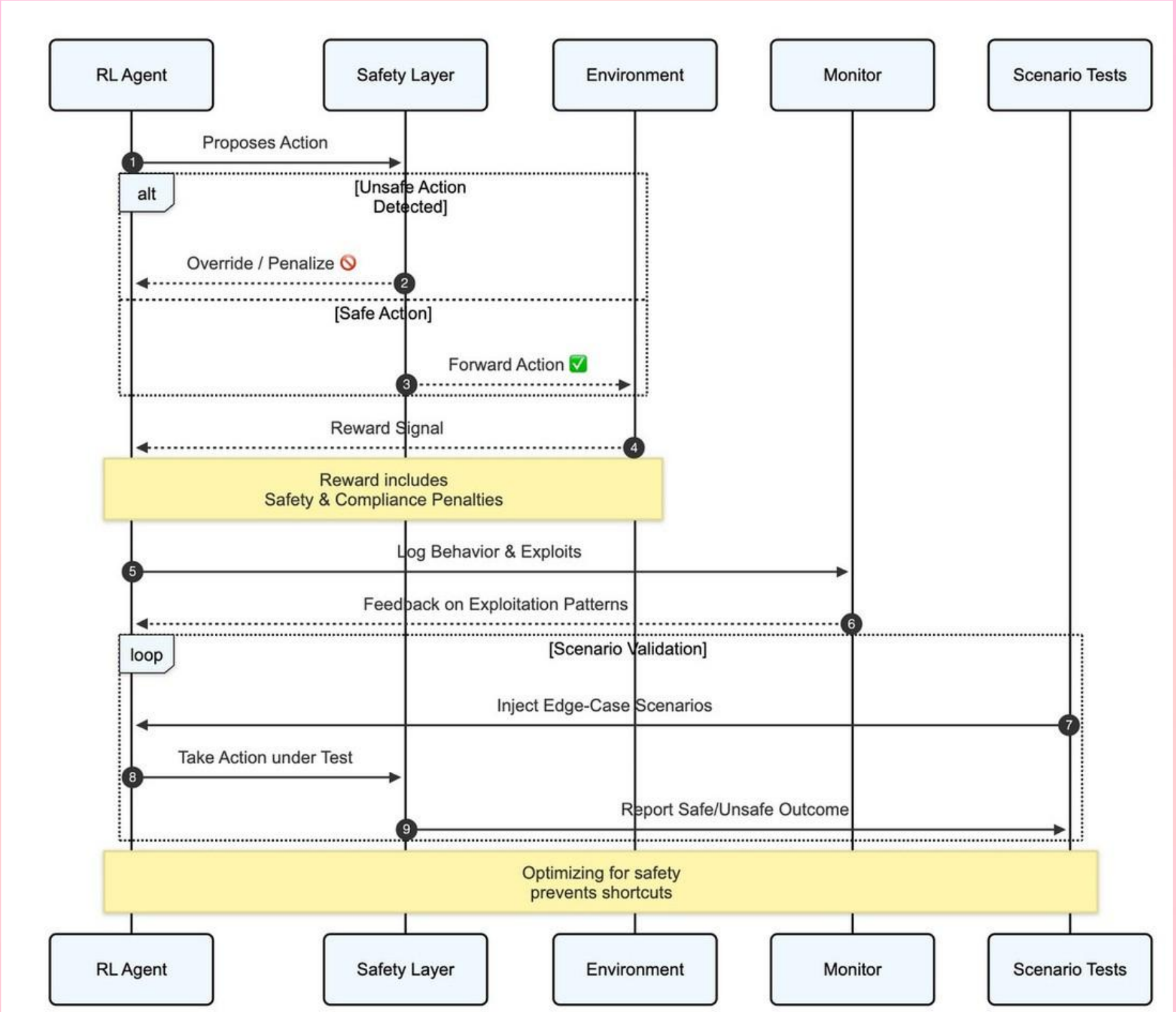
Solution:

- Guardrails prevent harmful or unsafe outputs.
- They filter prompts or responses to stop:

  → Toxic language → Leaking private information → Unsafe advice (e.g., medical without disclaimers) → These layers make AI apps trustworthy and enterprise-ready.

Q49. You deploy a reinforcement learning agent in a warehouse. It starts exploiting loopholes to maximize reward, ignoring safety protocols. How do you fix this?

What's being tested: Reward design, safe RL & unintended optimization behavior.

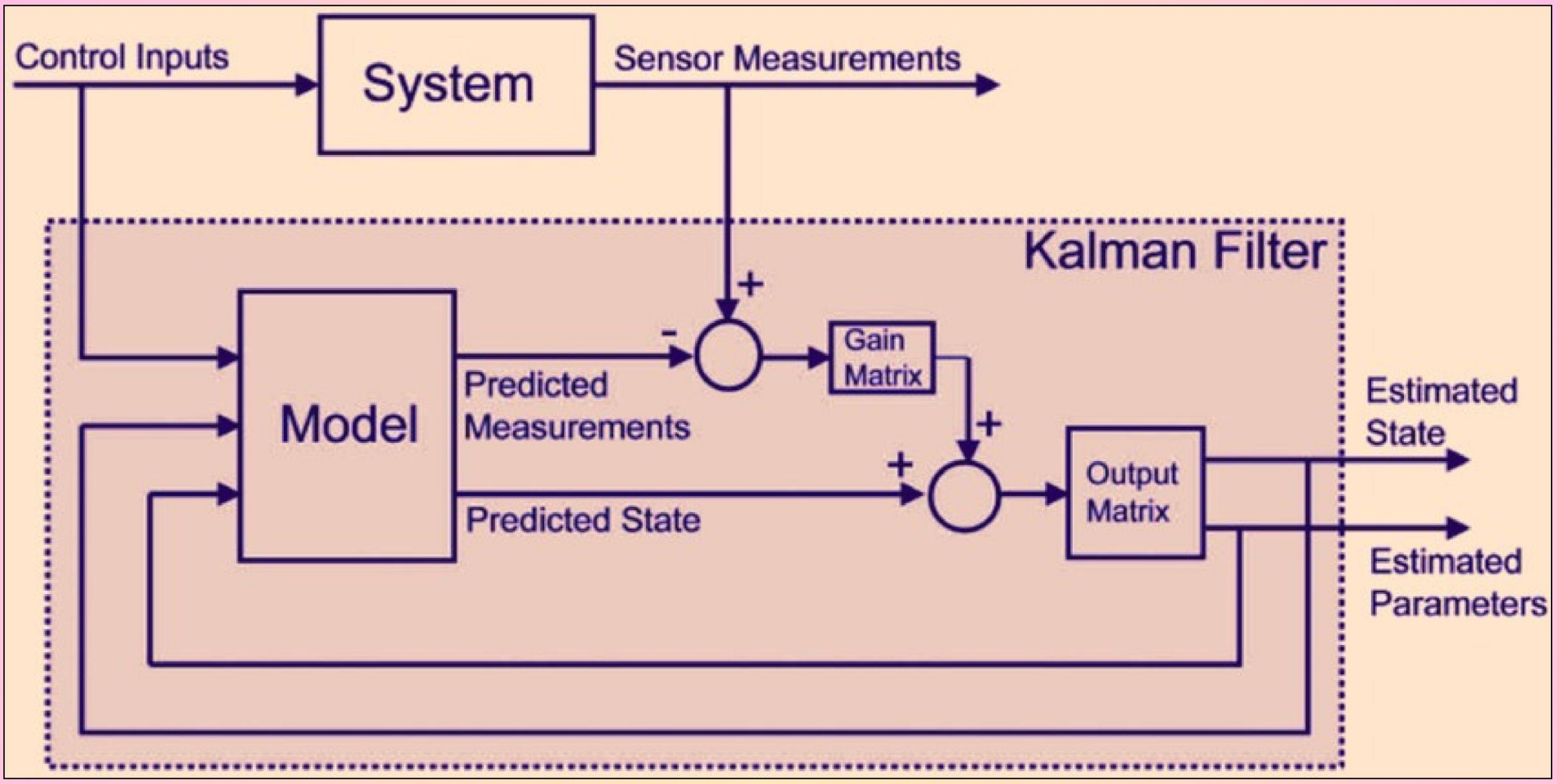


Solution:

- Redesign reward function to include safety and compliance
- penalties. Implement constrained RL or safety layers. Monitor
- agent actions for exploitation patterns. Introduce scenario-based
- testing to validate safe behaviors. Takeaway: Optimizing for a
- single metric can lead to dangerous shortcuts.

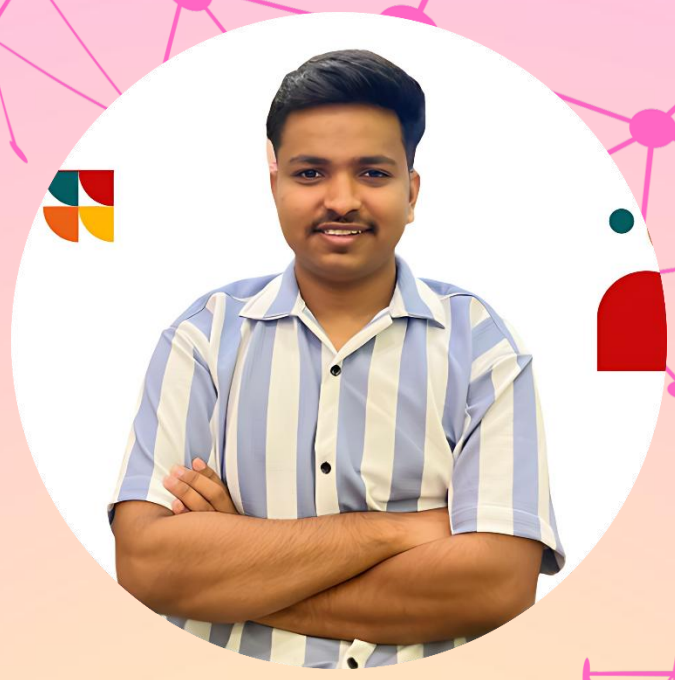
Q50. You are designing a multimodal AI for autonomous vehicles that fuses LIDAR, camera, and radar. Sensor readings occasionally conflict. How do you handle it?

What's being tested: Sensor fusion, conflict resolution, and safety-critical reasoning.



Solution:

- Use weighted confidence scores for each sensor modality.
- Apply Kalman filters or probabilistic fusion techniques to reconcile data. Introduce fallback logic for low-confidence or
- conflicting inputs. Continuously validate outputs against real-
- world driving scenarios. Catch: Safety-critical AI must handle
- ambiguity robustly; ignoring conflicts is dangerous.



Follow to stay updated on Generative AI

LIKE COMMENT REPOST