Agenda

→ Choosing the right LLM for our App^n.

→ Cost
- Model-A : 0.1 $ per million token
- Model-B : 10 $ per million token

→ Latency : Response time
- Very important parameter for Chat | Voice agents.
- Any real-time app^n.

→ Intelligence.
- Reasoning

⇒ While choosing the LLM for the use-case, we should try to find out the right balance b/w these 3 parameters.

Ex: Voice Assistant agent.

Latency - 800-900 ms (< 1sec)

Gemini flash 2.5.

- 250 tokens/sec
- 0.25 sec time-to-first-token.
- 0.15 $ / Million tokens.

GPT-4o.

- Better reasoning
- latency & cost will be on the higher side

Ex Chat Agent with complex reasoning

⇒ Claude Sonnet x

- Most predictable outputs ⇒ Accuracy ↑↑.
- Code Generation / structured output.

Use-cases: Legal analysis / financial analysis / ...

⇒ 15 $ / M tokens.

Ex: High Volume & Low Complexity Chat Agent.

Claude Sonnet ✗

Gemini flash 2.5.

- 250 tokens/sec
- 0.25 sec time-to-first-token.
- 0.15 $ / Million tokens.

100x Cheaper than Claude.

Model selection shouldn't be hard-coded.

⇒ We should be able switch over models based on the requirement.

⇒ Model selection should be configurable

1. Router Based Model selection.

→ Route the request to the right model based on the complexity of request.

Simple query ⇒ Gemini Flash
↳ fast + cheap.

Complex query ⇒ Claude Sonnet
↳ Smart + Consistent.

Image query ⇒ Gemini Pro, ...

2. Agent selection ⇒ Configurable.
↳ We'll be able to change the underlying model without making the code changes.

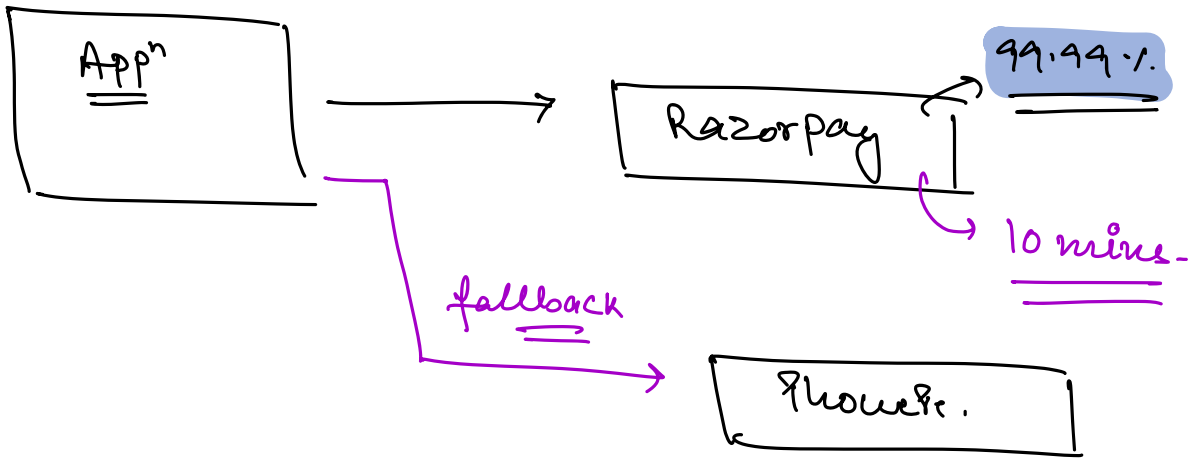
anthropic.messages.create(model = "claude-sonnet-4.5")
↳ Hard Coding

anthropic.messages.create(model = $model_config)
↳ Config variable

3. Fallback chain for Resilient systems.

SLA.
↳ Service Level Agreement

99.9%.



LLM Optimization Strategies.

1. Semantic Caching.
2. Prompt Optimization.
3. Batch Processing.
4. Two - Tier Processing