# Cost Considerations

## Usage costs will likely increase

- Multilayered architectures cost more than single-prompt systems. Each layer is an API call, and those add up. If you're running a four-layer cycle on every user interaction, you’re potentially paying 4x what a single prompt would cost (depending on usage costs and tokens). That's the trade you're making for better results

- But they work better when properly configured. The question isn't "should I add more layers to save money”; it's "does the quality improvement justify the cost for my use case?" A customer service bot might not need four layers. A narrative engine generating premium content probably does

- Optimization strategies exist. Use cheaper models for simpler layers (correction doesn't need the most expensive model), cache aggressively for cyclical systems, and be honest about cutting layers that aren't pulling their weight. Every layer should earn its spot

# Performance Considerations

## Speed vs quality

- More layers means more latency. If you need fast responses, waiting on three consecutive prompts to complete is probably a bad solution

- But parallelization can help. Some layers don't depend on each other and can run simultaneously. If your reasoning layer and your memory retrieval layer both only need the user input, run them in parallel

- Performance can be helped and hurt by layering. Adding a layer isn't always the answer. Sometimes consolidating two weak layers into one strong prompt improves both speed and quality