Think Architecturally

Strategize on the best ways to achieve great results for your platform

- Consider your actual goal and then break it down into steps. If you were to perform this action yourself, what steps would you need to follow. Write that down. That's your workflow

- After formalizing your workflow, think of the type of data transformations you would need throughout that process and then.build the prompts to automated, then chain them together

- Illustrative example: Your platform depending on conversation history for context can cause your token count and performance to take a hit. Perhaps a conversation consolidator prompt would benefit you. And if you want a truly random number to be used in the determination of something in your system, perhaps you have the backend serve that up to your AI, rather than assuming that the LLM's training data can produce anything close to pure randomness

Correction Layers

The referee of your platform

- Correction layers catch errors after other layers have done their work. They're your quality control layers. They spot continuity breaks, logical inconsistencies, or constraint violations that slipped through

- In Emstrata, the Chron-Con layer runs after the narrative is written. It checks for things like: Did a character who was in the tavern suddenly appear in the forest without traveling? Did someone use an item they don't have? Are the spatial coordinates consistent with the described action?

- **When you need one:** If there are complex requirements and expectations that your platform needs to meet. Correcting before revealing the final answer can lower the chance of bad responses