Cyclical Vs Circumstantial Systems

And everything in-between

- Cyclical systems run the same prompts every time. Emstrata follows this pattern: every turn runs Discovery, then Narration, then Chron-Con, then Groundskeeper, in that exact order. The flow is predictable and consistent regardless of what happens in the simulation. You always know what's executing next, which makes debugging straightforward and cost estimation more reliable

- Circumstantial systems determine the pathway based on outcomes or AI direction. The route through your architecture changes depending on what happened in previous steps. Maybe an error detection layer decides whether correction is needed. Maybe a routing layer examines user intent and sends the request down completely different processing paths. The system adapts its own execution flow based on runtime conditions

- Hybrid systems are mostly cyclical at their base, but circumstantial at times when specific conditions warrant different handling. You might always run your core cycle, but branch to specialized subsystems when certain triggers fire. Many real-world systems end up here. It’s a reliable backbone with conditional branches for edge cases. Emstrata has a number of circumstantial offshoots as well

Agnostic Backend Interaction

What happens between AI layers

- Between AI layers, it's important to save important, transformed data to the backend for future retrieval, debugging, rerunning if there's an error, etc.
- Saving data also allows you to present that data in interesting ways later or feed that data into other layers in the future
- Also, when you need an unbiased judge, the backend is the place to go. The backend is 'agnostic' to outcome, whereas the AI may or may not have a strong preference and display it
- In Emstrata, consequences are rolled and use weighted randomness. The Discovery layer determines the likelihood something happens, and then the backend returns a random number out of 1000. If that number is within than the set likelihood range, the backend serves the confirmed consequence to the Narration layer, if it's outside of the range, it sends the failure outcome