

Five Layers. One Missing Owner.

Azure Copilot Studio Governance

Three enterprises audited. All had Copilot Studio active in production. None could articulate their system prompt governance model, version control, data lineage, or incident response plan. Copilot Studio governance is a control architecture problem — five distinct layers, each with its own audit surface.



Deployment configuration. What data sources and connectors does this Copilot access, and is that access scoped to what the use case needs?



System prompt governance. Who can change the system prompt, and what's the review process? A prompt change is a behavior change.



Data lineage and training. What data trains or grounds this agent's responses, and can you produce that lineage on demand?



Change management and versioning. Is there a version history, and can you roll back to a known-good state?



Incident response protocol. If the agent gives harmful output, who gets notified, how fast, and what's the containment step?

THE BOTTOM LINE

Layer two is the most commonly undocumented, because it feels informal. A prompt change can alter what an agent tells a customer — that's a behavior change with real downstream consequences.

Inventory every deployment against these five layers — solutionsbyjewel.com