TechVision's AI Agent Risk Map

This one-page risk map is designed to help enterprise teams quickly identify where AI agents and copilots create meaningful operational, security, and compliance exposure, and where a runtime governance layer such as SafePrompts.ai can reduce that exposure.

TVR Labs positions SafePrompts.ai as an AI governance control plane that governs prompts, tools, models, and actions before, during, and after execution, including identity-aware policy checks, real-time risk scoring, throttling unsafe actions, approval routing, and auditability.

Risk Map

Agent / Copilot Type	Data leakage	Destructive actions	Compliance breach	Hallucinated or unsafe decisions	Tool / supply-chain abuse
Coding agent	Source code, secrets, or environment tokens can leak through prompts, files, logs, or connected tools if the agent can access more context than it needs.	A coding agent can delete production databases, volumes, or backups if it inherits broad credentials and no platform-level approval gate blocks destructive commands.	If the agent touches production systems, customer data, or regulated environments without enforced policy boundaries, it can create audit and governance failures even when acting "helpfully."	The agent may guess instead of verify, apply the wrong fix, or execute a risky change because model reasoning is probabilistic and prompts alone do not create hard boundaries.	The agent can misuse CI/CD, infrastructure APIs, package managers, or deployment tools when permissions and action scopes are broader than the task requires.
Customer support copilot	A copilot can surface confidential emails, internal notes, or sensitive account details in a multi-turn thread when retrieval scope exceeds what the support context actually requires.	Direct destructive risk is lower, but the copilot can trigger harmful account actions, send incorrect responses, or expose information that causes downstream customer harm.	Confidential content may be summarized despite policy expectations, as seen in Microsoft's confirmed Copilot bug involving confidential emails.	The copilot can generate confident but wrong answers about account status, entitlements, or policy, leading agents to make customer-facing mistakes at speed.	Integrated CRM, email, knowledge base, and ticketing connectors expand the blast radius if the copilot can query systems or pass data across tools without guardrails.
Internal workflow orchestrator	Cross-functional workflow agents can aggregate sensitive HR, finance, legal, or customer data that no single user intended to expose in one place.	Orchestrators can approve, route, modify, or trigger high-impact workflow steps automatically, creating large downstream effects if controls are missing.	When approvals, retention rules, or segregation-of-duties checks live outside the workflow loop, the agent can violate internal policy while appearing efficient.	These agents can misclassify requests, route work incorrectly, or apply incomplete logic across complex business processes if no review threshold is enforced.	Workflow tools, SaaS connectors, and automation platforms can be chained together in ways that amplify a single bad decision across many systems.

Procurement agent	The agent may expose sensitive pricing, contract terms, supplier assessments, or internal negotiation notes to the wrong users or systems.	A procurement agent can auto-approve risky spend, contract renewals, or onboarding actions that later become difficult to reverse.	If sanctions, security, legal, or privacy checks sit outside the agent's approval path, contracts can move forward without required review.	The agent may treat incomplete vendor data as sufficient, misread non-standard clauses, or over-trust templates and benchmarks in edge cases.	Supplier platforms, contracting tools, and third-party risk systems create a supply-chain surface where weak connector governance can propagate bad approvals.

Control Patterns

Risk type	Governance pattern at the prompt / agent layer
Data leakage	Classify sensitivity before model access, minimize retrieval scope, redact or block sensitive categories, and enforce identity-aware prompt and response policy at runtime.
Destructive actions	Put destructive tools behind hard policy gates, require explicit approval for irreversible actions, use least-privilege credentials, and separate production from staging access paths.
Compliance breach	Route regulated or high-risk interactions through policy checks tied to role, data class, and action type, then log decisions for audit and review.
Hallucinated or unsafe decisions	Treat high-impact outputs as proposals, require review thresholds, score risk in real time, and constrain tool use when confidence, context, or policy conditions are weak.
Tool / supply-chain abuse	Govern which tools an agent may call, under what circumstances, with what parameter constraints, and with full traceability across connected systems.

Design Notes

The practical lesson from recent incidents is that rules written in prompts or configuration files are not enough on their own; the safer pattern is to encode boundaries in tooling that can physically block, throttle, or escalate risky actions.

For teams evaluating governed agents, the first useful question is not "Which model should be used?" but "What identities, data, tools, and irreversible actions should this agent be allowed to touch without additional approval?"