# Risks and Assumptions – Agentic AI Observability

## Risks

|  |  |  |
| --- | --- | --- |
| Risk | Description | Mitigation Strategy |
| Model Accuracy Drift | ML models may lose accuracy over time due to changes in application behavior or infrastructure | Schedule regular retraining; monitor model performance metrics |
| Data Gaps or Latency | Gaps in Prometheus, AppDynamics, or Splunk data can reduce agent reliability | Implement a validation agent; include fallback logic |
| API Rate Limits / Throttling | AppDynamics and other tools may throttle excessive API usage | Use parallelism with timers; apply exponential backoff + caching |
| False Positives in Automation | Automated remediation could trigger on incorrect signals | Require human approval or validate before action is applied |
| Security Risks in Open Interfaces | APIs without auth could allow misuse | Apply RBAC and audit logging |
| LLM Interpretation Misalignment | LangChain/LLM could misinterpret prompts | Use prompt validation and intent classification |
| Tooling or Integration Failure | If Prometheus/AppD/Splunk fail, agents break | Add health checks and fallback logic |

## Assumptions

|  |  |
| --- | --- |
| Assumption | Justification |
| Limited Scope During POC | System is trained on a small set of applications and short time window to validate feasibility |
| Metric Paths Are Known and Stable | Paths from Prometheus/AppDynamics remain consistent for model and rule usage |
| User Input Is Intent-Aligned | Users will query with supported prompt types; unsupported queries handled by validator |
| Data Is Fresh and Streaming Regularly | Metrics/logs are assumed to flow in real time |
| Limited Concurrent Users During POC | Chat + agent workload expected to be single-user or low concurrency |
| Pre-trained Models Are Available | Agents rely on access to trained model files via model store |
| Automation Actions Are Controlled | Actions are reviewed or sandboxed before being triggered automatically |