🧭 Service Blueprint – Red Hat OpenShift AI v3 (RHOAI UI v3)

Enabling a Secure, Guided, and Scalable GenAI Platform for AI Engineers and Platform Engineers

---

📌 1. Customer Actions

The steps users take to interact with the system.

Stage	AI Engineer Journey	Platform Engineer Journey
Discover / Ideate	Search for GenAI use case inspiration → Use Gallery & QuickStart Wizard → Book intro call	Define project policy → Configure catalogs and auth levels → Setup GPU/project resources
Prototype / Build	Use Dev Sandbox Wizard → Connect data → Test models/tools → Setup prompt/guardrails	Enable MCP, MaaS, Agent-as-a-Service on infra → Define approval workflows & usage quotas
Evaluate / Iterate	Run Auto-Evals & prompt iterations in Playground → Export to code	Review evaluations and logs → Validate metrics and usage alignment
Deploy / Share	Push agent to app integration → Submit for approval → Share internally or externally	Approve deployment → Monitor usage/security/compliance → Optimize infra allocation

---

🧰 2. Frontstage (Visible Touchpoints)

UI / Interface	Description
RHOAI UI v3 Gallery	Guided exploration of use cases and starting points for GenAI apps
Dev Sandbox Wizard	Environment provisioning with presets and data connectors
Inference Playground	Compare models, test prompts/tools, tweak RAG setup, run evals, add guardrails, export to code
Model / MCP Registry	Unified catalog for discovering, filtering, and launching models and tools
Eval & Guardrail Console	Interface to view safety eval results, policy enforcement metrics, and risk grading
Approval Panel (PEs)	Dashboard for platform engineers to approve or reject AI asset configurations and usage
Dashboards & Logs	SLI/SLO monitoring, token usage views, drift detection, trace logs

---

⚙️ 3. Backstage Actions (Internal)

Team	Action
Platform Engineering	Maintain catalogs, model/card curation, enable RBAC & secrets integration, set up GPU inference nodes
DevOps / MLOps	Configure Llama Stack / vLLM infra, manage CI/CD pipeline examples for agent deployment
UX / Product Team	Create Kickstarts, sample templates, onboarding wizards, walkthrough videos
Security / Compliance	Define sandboxing rules, audit policies, legal approvals for deployment

🧱 4. Supporting Processes / Infrastructure

System / Component	Role in Support
Model Registry & Catalog	Source of truth for models, with usage metadata, tagging, and context
MCP Registry & Tool Store	Policy-aware storage for tool-calling libraries, data connectors, and agent utilities
Llama Stack + vLLM	Shared inference serving platform with GPU routing and sandbox execution
GitHub + GitOps	Dev integration pipeline for agent apps, eval automation, versioning
Trusty AI Eval Framework	Risk scoring, model grading, and human-in-the-loop enforcement
AuthZ & RBAC Engine	Project-level and user-level access control, integrated with OpenShift cluster policies
Logging + Observability Stack	Token cost tracing, response logging, drift tracking, alerting

---

🎯 Key Takeaways from Research

- Users want **rapid feedback loops** and tools to move from idea to deployed agent quickly
- The "Blank Page Problem" is real → **guided starting points** and golden paths are vital
- Platform Engineers demand **governance-first architecture**: validated catalogs, strict RBAC, and auditability
- AI Engineers want to **stay in their IDE**, using modern frameworks (LangChain, LangGraph, etc.) with a clear “sandbox → prod” workflow