Case Study: “BrokerBot AI” at GlobalRe Brokerage

Background
GlobalRe Brokerage, a top-tier insurance & reinsurance broker, has developed **BrokerBot AI**—a Gen AI platform that automates treaty structuring, client risk assessments, and real-time inquiry handling via autonomous agents. After a successful pilot with two major cedents, the firm now needs to scale BrokerBot AI across its entire service portfolio and deploy agents into production under rigorous risk controls.

---

1. Portfolio Execution for BrokerBot AI

- **Stage-Gate Pipeline**
  Transition each BrokerBot use case through:
  1. **Pilot:** Proof-of-concept on select treaty deals (accuracy ≥ 90 % on pricing).
  2. **MVP:** Client-facing demo, user-acceptance ≥ 80 %.
  3. **Staging:** Parallel run alongside human brokers, ROI ≥ 15 %.
  4. **Production:** Full rollout across all desks.
  5. **Retire:** Sunset underperforming models.
- **Impact-Risk Prioritization**
  Score initiatives by revenue potential (treaty volume), technical complexity (data quality), and compliance risk (jurisdictional regs). Fund top-quartile projects first.
- **Cross-Functional Squads**
  Form “Value Teams” pairing Product Managers, Actuaries, Data Engineers, ML Engineers, Security & Legal—each squad owns its BrokerBot module end-to-end.
- **Iterative Feedback Loops**
  At every gate, run A/B tests on pricing recommendations, gather broker surveys, and perform performance retrospectives—iterate using real claim-loss data.
- **Governance & Funding**
  A quarterly **AI Steering Committee** reviews a live Portfolio Dashboard showing status, burn rate, technical debt, and dependency maps—reallocating budget and retiring low-performers.

---

2. Autonomous Agent Architecture & Deployment

- **Modular “Core + Skills”**
  BrokerBot agents share a core LLM kernel plus plug-ins (“skills”) for treaty parsing, risk-factor extraction, client-Q&A, and regulatory-check modules—each versioned independently.
- **Scalable Orchestration**
  Agents run on Kubernetes with event-driven queues: claim-notification triggers spin up pricing agents; incoming client emails queue the Q&A skill; autoscaling reacts to workload.

- **Safety-First Guardrails**
  Runtime checks enforce permission gates (no unauthorized data writes), rate-limits on treaty modifications, and “red-button” human-in-the-loop for high-risk actions (e.g., treaty cancellation).
- **Continuous Learning & Auto-Tuning**
  Deploy new agent builds via canary releases; shadow-mode collects feedback on live treaty outcomes; auto-tune models monthly on newly labeled claim data.
- **Observability & Incident Response**
  Centralized logging (latency, error rates, drift metrics) with anomaly alerts; SLAs ensure on-call response within 1 h for critical failures.
- **Lifecycle & Version Management**
  Semantic versioning, blue/green deployments, and deprecation schedules with rollback playbooks guarantee safe updates.

---

3. Bridging Pilot to Production

- **Maturity Assessments**
  Before each stage-gate, score BrokerBot modules against production-readiness criteria: data quality, compliance sign-off, performance benchmarks.
- **Runbooks & SLAs**
  Standardize deployment checklists: rollback procedures, incident-response steps, support tiers and uptime guarantees (99.9 %).
- **Infrastructure as Code**
  Manage staging and prod environments declaratively via Terraform—ensuring reproducibility and auditability.
- **Change Management**
  Conduct broker training sessions, publish SOP updates, and include “What’s Changed” notes with every release.

Participant Exercises

Discussion Questions

1. **Prioritization Challenge:**
   How would you adjust the Impact-Risk scoring if a new data-privacy regulation raises compliance risk?
2. **Agent Safety:**
   Propose one additional runtime guardrail for the Q&A agent that handles client treaty inquiries.
3. **Governance Gaps:**
   Identify a missing KPI on the Portfolio Dashboard that could expose hidden technical debt.
4. **Change Management:**
   What key elements must be in the “What’s Changed” release notes to minimize broker resistance?

Key Deliverables

1. **Stage-Gate Criteria Matrix:**
   A table listing gates, entry/exit thresholds, and responsible roles.
2. **Value-Team Org Chart:**
   Diagram of squad composition and end-to-end ownership for each BrokerBot module.
3. **Agent Architecture Diagram:**
   Visual of core + skills plug-ins, orchestration flow, and safety guardrails.
4. **Runbook Outline:**
   Checklist for deployment, rollback, incident-response steps, and SLA definitions.
5. **Portfolio Dashboard Mock-Up:**
   Live-data wireframe showing status, burn rate, technical debt, dependencies, and custom KPI you recommend.

Map each deliverable back to the corresponding branch of the mind-map and be prepared to justify how it ensures a smooth, safe, and scalable transition from pilot to production.