Incident Management – Agentic AI Demo Script

Ticket Creation & Escalation

Frame	Visual	Talk Track	Notes
1	Ticketing system dashboard showing a new ticket. The ticket was created either via automated fault detection (syslogs, alarms) or from an end-customer call. NOC L1 engineer clicks "Escalate to Cisco AI".	In a real-world operations environment, incidents come in from multiple sources — automated fault detection picking up syslogs and alarms, or directly from an end-customer call reporting an issue. A ticket has been created in the ticketing system and is now visible to the NOC team. The NOC L1 engineer reviews the ticket and decides to escalate it to Cisco AI for investigation. [clicks "Escalate to Cisco AI"]	**Ticket shown in ticketing system:** "Ticket ID: . Source: . Description: . Severity: . Status: Open. Relevant logs/ fault snippets attached."

Initial Interaction & Investigation Trigger

Frame	Visual	Talk Track	Notes
2	IO Agent UI opens with the ticket context loaded. IO Agent displays the ticket summary and asks targeted follow-up questions. **Card-1:** Ticket details card showing problem statement, source, severity, and attached logs.	The IO Agent picks up the ticket and reviews the context — the problem statement, logs, and any fault snippets. If it needs more information, it asks targeted follow-up questions to the engineer. Once it has enough context, it automatically triggers the investigation workflow.	**IO Agent:** "Ticket received. Reviewing problem context. I can see . Can you confirm ?" After response: "Sufficient context collected. Triggering investigation workflow."
3	IO Agent shows a status indicator: "Investigation in progress." High-level orchestration logs scroll on screen. **Card-2:** Orchestration status card showing problem categorization, intent mapping, and use case identification.	Behind the scenes, the IO Agent is categorizing the problem, understanding the customer's intent, and mapping it to relevant use cases. It has decided to invoke the Master Reasoning Agent for deep analysis. You can see the high-level orchestration logs here.	**IO Agent Thinking:** "Categorizing problem: . Mapping customer intent. Identifying relevant use cases. Invoking Master Reasoning Agent (MRA) for investigation."

MRA Investigation & Specialized Troubleshooting

Frame	Visual	Talk Track	Notes
4	MRA progress logs visible on UI. **Card-3:** Data acquisition summary showing sources queried — Knowledge Graph, RAG, topology, device states, protocols, telemetry, alternate paths.	The Master Reasoning Agent is now gathering data from multiple sources — our Knowledge Graph, retrieval-augmented generation for relevant documentation, network topology, device states, protocol information, telemetry data, and alternate paths. It's building a comprehensive picture of the network state.	**MRA Logs:** "Querying Knowledge Graph. Fetching topology data. Retrieving device states for . Pulling telemetry data. Analyzing protocol states. Checking alternate paths. Data acquisition complete."
5	MRA deployment logs visible. **Card-4:** Specialized agents deployment card showing which agents (BGP, IGP, MPLS, SR, L2/L3, etc.) have been activated and their status.	Based on the data gathered, the MRA has determined which specialized agents are needed. In this case, it's deploying agents to perform deep domain-specific troubleshooting. Each agent is an expert in its domain — they know exactly what to look for.	**MRA Logs:** "Analyzing collected data. Determining required specialized agents. Deploying BGP Agent. Deploying IGP Agent. Deploying Agent. Agents deployed — awaiting findings."
6			**Agent Findings:** "BGP Agent: . IGP Agent: . MPLS Agent: . All findings returned to MRA for correlation."

Frame	Visual	Talk Track	Notes
	Specialized agents return findings. **Card-5:** Troubleshooting findings card showing each agent's results, correlated symptoms, and evidence trail.	The specialized agents have completed their analysis and sent their findings back to the MRA. Each agent has examined the problem from its domain-specific perspective — checking protocol states, adjacencies, route tables, label paths, and more.	

Root Cause Analysis

Frame	Visual	Talk Track	Notes
7	MRA synthesizes root cause. **Card-6:** Root cause analysis card showing: root cause description, affected components, evidence chain, and confidence score. **Card-7:** Topology view highlighting affected nodes/links and the fault location.	This is where the magic happens. The MRA correlates all the inputs from the specialized agents and identifies the precise root cause with supporting evidence. It provides a confidence score so the engineer knows how certain the analysis is. You can see the affected components highlighted in the topology view.	**MRA:** "Correlating findings from all specialized agents. Root cause identified: . Affected components: . Evidence: . Confidence score: ."
8	IO Agent displays executive-friendly summary. **Card-8:** Executive summary card — What happened, Why it happened, Where it happened, Business impact.	The MRA hands over its findings to the IO Agent, which presents them in a clear, executive-friendly format. No need to dig through logs — you can immediately see what happened, why, where, and what the business impact is. This is the kind of summary you can share with management or the customer.	**IO Agent Summary:** "What: . Why: . Where: . Business Impact: ."
9	IO Agent generates a downloadable PDF. **Card-9:** RCA report card with a download button and preview of the report contents.	The IO Agent has also generated a detailed RCA report as a downloadable PDF. This can be attached to the ticket, shared with stakeholders, or archived for compliance. Everything is documented — the timeline, analysis steps, evidence, and conclusions.	**IO Agent:** "RCA report generated. Report includes: incident timeline, data sources queried, agent findings, root cause analysis, evidence, and recommendations. Available for download."

Remediation

Frame	Visual	Talk Track	Notes
10	IO Agent prompts the user: "Would you like to check feasibility for remediation via configuration changes?" User clicks "Yes".	Now that we know the root cause, the IO Agent asks whether we'd like to explore remediation options. Can this be fixed with a configuration change, or does it require physical intervention? Let's find out. [clicks "Yes"]	
11	MRA evaluates feasibility. **Card-10:** Feasibility assessment card showing the decision — configurable fix or physical intervention required.	The MRA has evaluated the remediation feasibility. In this case, it has determined that the issue can be resolved via a configuration change — no physical intervention needed. It communicates this decision back to the IO Agent with full justification.	**MRA:** "Evaluating remediation feasibility. Analyzing root cause against configuration capabilities. Decision: Issue can be remediated via configuration change. Justification: ."
12	IO Agent invokes the Intent-Driven Configuration Agent. **Card-11:** Proposed configuration card showing the generated configuration with highlighted changes.	Since a configuration fix is feasible, the IO Agent invokes our Intent-Driven Configuration Agent. This agent takes the intent, constraints, and policies into account and generates the proposed configuration. You can review exactly what will be changed right here.	**IO Agent:** "Invoking Intent-Driven Configuration Agent. Intent: . Constraints: . Policies: ." **Config Agent:** "Configuration generated. Displaying proposed changes."
13		This is a critical human-in-the-loop checkpoint. The engineer reviews the proposed configuration before anything is deployed. Nothing goes to the network without	

Frame	Visual	Talk Track	Notes
	Human approval checkpoint. UI shows the proposed configuration with "Approve" and "Reject" buttons. User clicks "Approve".	explicit human approval. The configuration looks good — let's approve it. [clicks "Approve"]	
14	Dry-run execution via NSO. **Card-12:** Dry-run results card showing validation outcome — commit check passed, no conflicts detected.	Before pushing to the live network, we execute a dry run using NSO. This validates the configuration against the current network state — checking for conflicts, syntax issues, and policy violations. The dry run has passed successfully.	**NSO Dry Run:** "Executing dry run. Validating configuration against network state. Commit check: Passed. Conflicts: None. Policy violations: None. Dry run successful."
15	Configuration deployed via NSO with live status updates. **Card-13:** Deployment status card showing real-time progress — device-by-device deployment status.	The dry run passed, so now we push the configuration to the network using NSO. You can see the live deployment status updating in real time — device by device. The configuration has been successfully applied.	**NSO Deployment:** "Pushing configuration to network. Device : Configuration applied. Deployment complete. All changes committed successfully."

Verification & Ticket Closure

Frame	Visual	Talk Track	Notes
16	Post-deployment verification completes. **Card-14:** Verification card showing service recovery confirmation — metrics back to normal, health checks passed. Ticketing system shows ticket updated with RCA report attached and status set to "Close Pending".	Post-deployment verification confirms the fix is working — service metrics are back to normal and all health checks have passed. The AI automatically updates the ticket in the ticketing system, attaches the full RCA report, and sets the status to "close pending" — ready for final review and sign-off by the operations team. From detection to resolution, all handled with Agentic AI.	**IO Agent:** "Post-deployment verification complete. Service metrics restored to baseline. Health checks: Passed. Updating ticketing system. Ticket : RCA summary added. RCA report attached. Status set to: Close Pending."

Cards Summary

Card	Content
Card-1	Ticket details — problem statement, source, severity, attached logs
Card-2	Orchestration status — problem categorization, intent mapping, use case identification
Card-3	Data acquisition summary — sources queried (Knowledge Graph, RAG, topology, telemetry, etc.)
Card-4	Specialized agents deployment — agents activated and their status
Card-5	Troubleshooting findings — each agent's results, correlated symptoms, evidence trail
Card-6	Root cause analysis — root cause, affected components, evidence chain, confidence score
Card-7	Topology view — affected nodes/links and fault location highlighted
Card-8	Executive summary — what, why, where, and business impact

Card	Content
Card-9	RCA report — downloadable PDF with full incident documentation
Card-10	Feasibility assessment — configurable fix or physical intervention decision
Card-11	Proposed configuration — generated config with highlighted changes
Card-12	Dry-run results — validation outcome from NSO
Card-13	Deployment status — real-time device-by-device deployment progress
Card-14	Verification & closure — service recovery confirmation, ticket updated to "Close Pending"