Agentic AI Troubleshooting across multi vendor



Oscar Toorell
Director Cloud RAN Technology & Ecosystem
Ericsson

Agenda

- Activity background
- Use case description
- What is an agent
- PoC architecture
- Role of agents in troubleshooting flow
- Agent orchestration
- Key POC takeaways

About this activity

- For any major event impacting RAN service, Field Service Engineers (FSE) will be deployed to site. Any site visit requires substantial cost and time.
- With multiple parties involved in a disaggregated solution, it is a risk new interfaces, handling and supplier support will create additional site visits
- The agentic AI troubleshooting agent PoC for Cloud RAN on RedHat Open Stack platform explored feasibility to localize fault root cause remotely and propose remote remedies where applicable.
- The PoC focused on exploring how knowledge from multiple vendors can be aggregated and utilized to provide insights and solutions in a multivendor deployment scenario.
- An Agentic AI based solution was selected in order to be able to work on the several instances autonomously



Definition of Agent

1	2	3	4	5
Acts autonomously	**Perceives environment**	**Processes information**	**Adapts with memory**	**Produces actions**
authorized to act, decide, and self-initiate tasks independently on behalf of a person or entity based on input goal or intent	uses external entities like data streams or other agents	applies rules, program logic or AI models to that data	retained, dynamic adaptation to changing conditions	further actions and execution to achieve goals autonomously or with a human intervention

Mechanisms needed for an agent



Guided by goals
(input, symptom)

- Prompted by end user
- Trigger input (alarm or event)
- Triggered by another agent



Perceives its environment

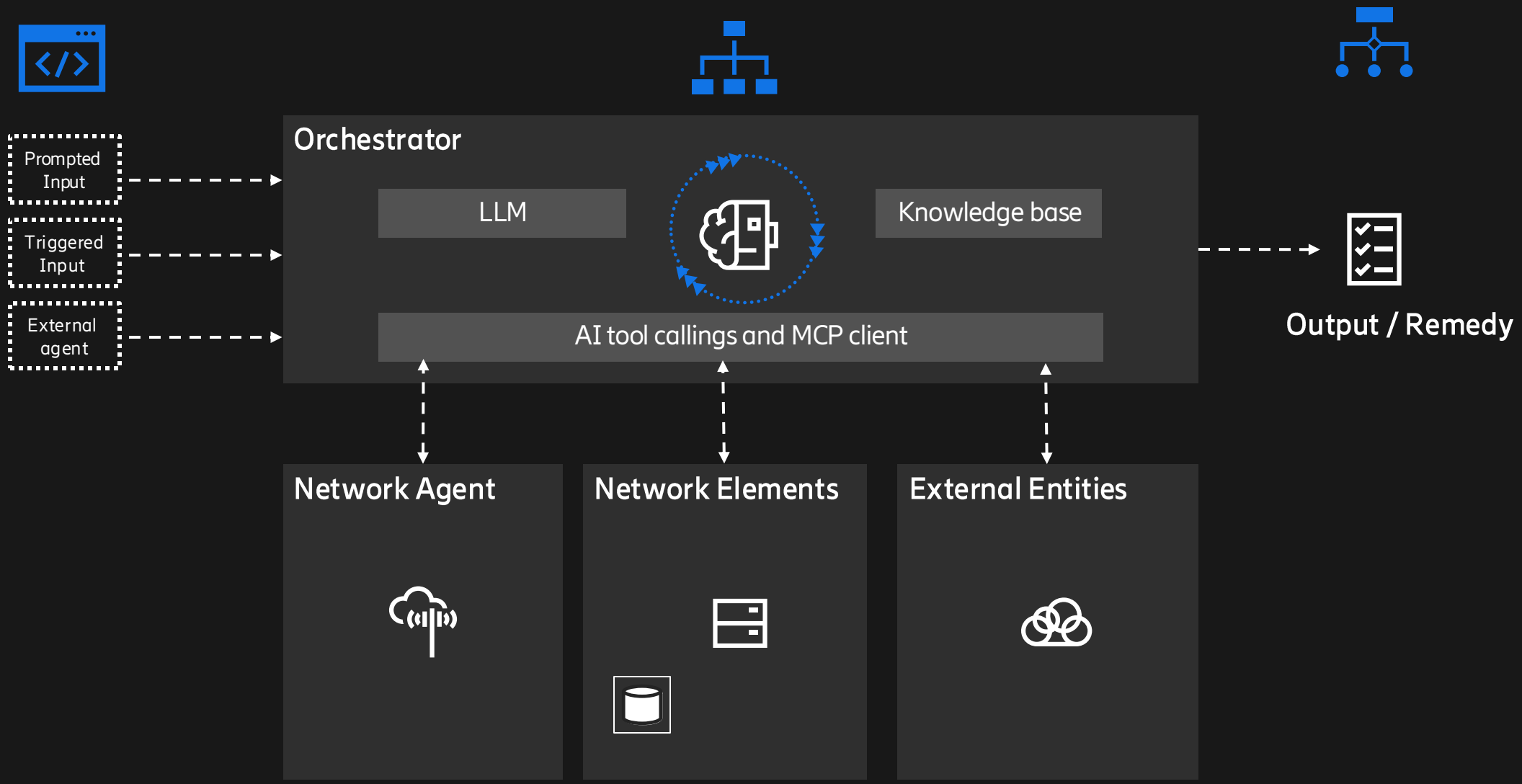
- External entities through MCP architecture (chat bots, data bases etc)
- More advanced remote agents through A2A architecture



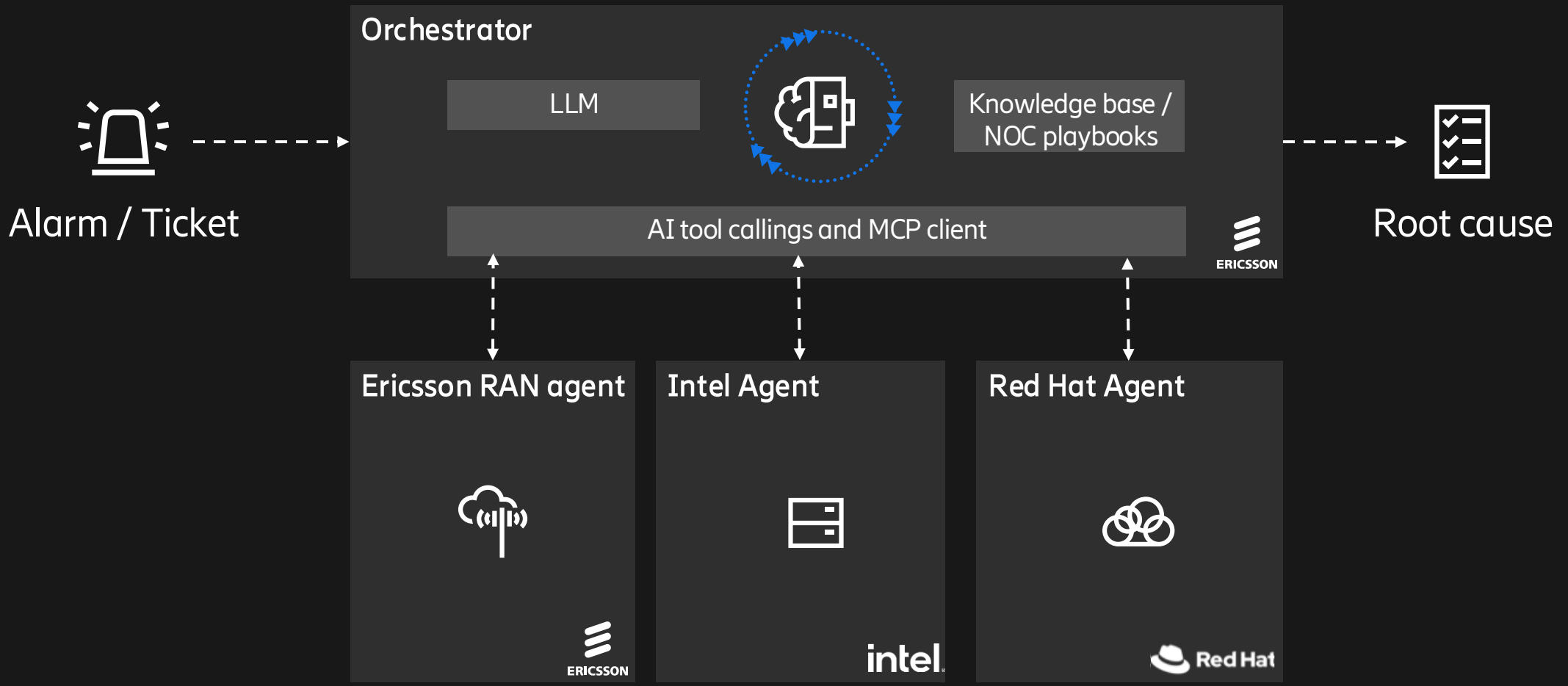
Process the received information and Produce results (output, remedy)

- Realized by different flavors of agentic orchestrator architecture and reasoning logic and knowledge bases

AI Agent conceptual view

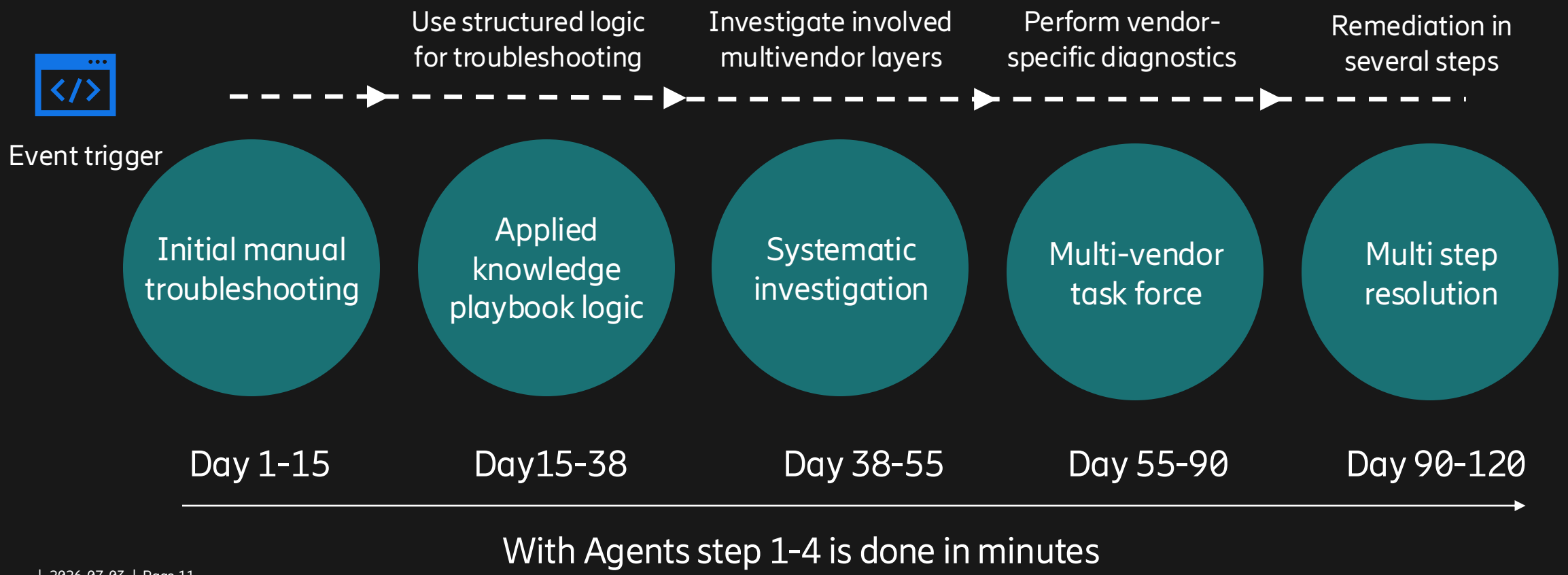


Multivendor agentic system architecture PoC

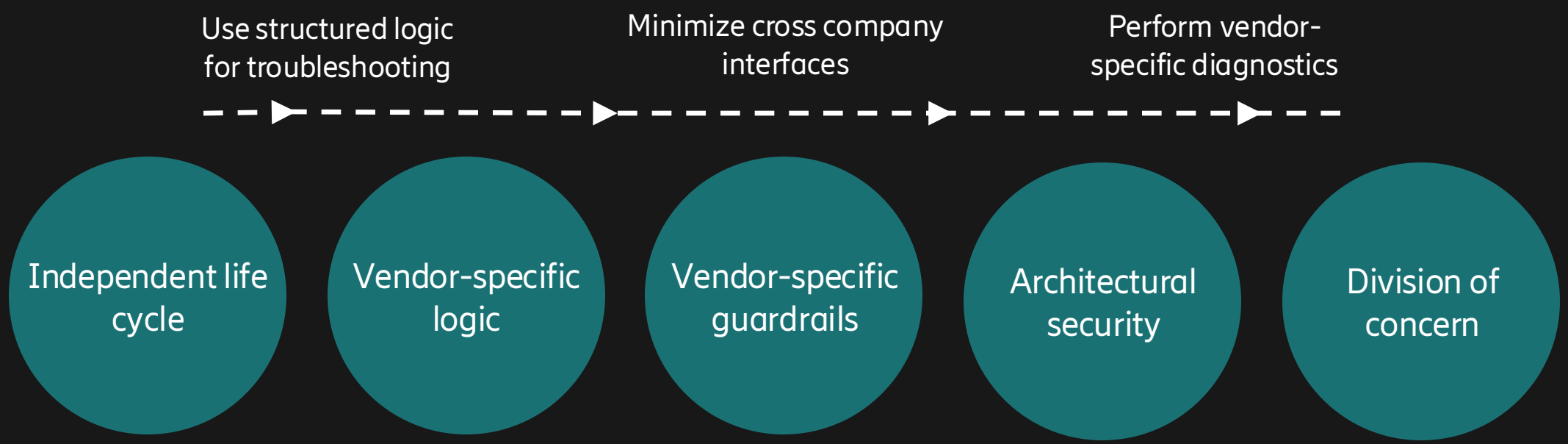


Architecture used in the PoC

Example end-to-end troubleshooting of multivendor Cloud RAN deployments



AI agents efficiently address key concerns for multivendor knowledge integration



Roles of the components used in PoC architecture

Solution Plan Agent (Orchestrator)	Initiates diagnostics based on detected impairments, determines required agents, sequences troubleshooting actions based on Ericsson knowledge base. Leverages internal services to connect to the Ericsson RAN Agent, Red Hat OpenShift MCP server, and Intel Accelerator MCP server etc., and determines action based on responses and playbook
Ericsson RAN Agent	Performs Cloud RAN log analytics, FM alarms, PM counter analysis, CM configuration states analysis, and providing actionable RCA insights based on internal tickets knowledge base
Red Hat Agent (MCP Endpoint)	Executes virtualization platform health checks, including node status, pod health, and resource utilization, looks up relevant documentation
Intel Agent (MCP Endpoint)	Performs NIC timestamping and hardware timing checks to identify accelerator-layer performance issues

Agent Orchestrator Examples

- **Rule based orchestrator**

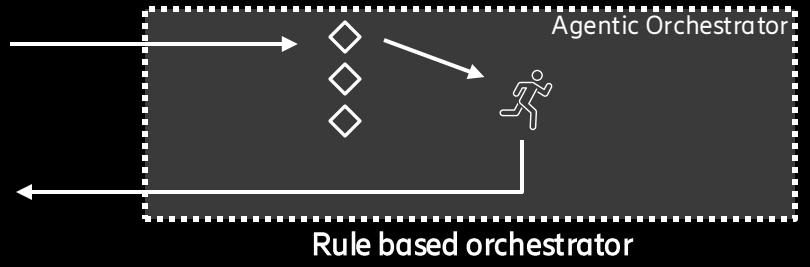
  End user selects and executes pre-defined workflow

- **Basic AI orchestrator**

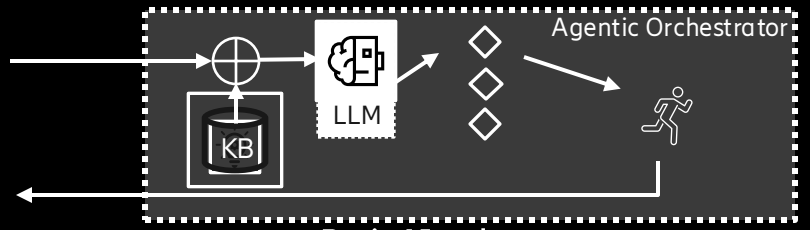
  Orchestrator finds and executes the most suitable pre-defined workflow based on input symptom

- **Advanced AI orchestrator**

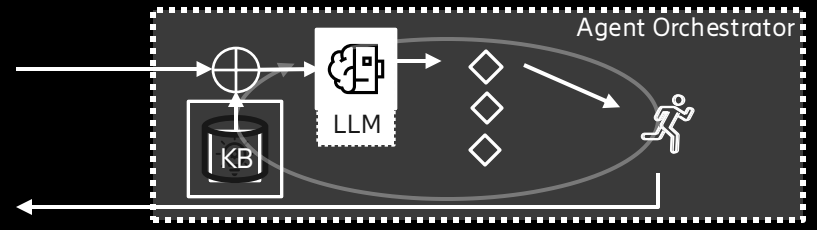
  Orchestrator finds and executes the most suitable pre-defined workflow based on input symptom, invoking external parts of the system and learning from the results of the analysis



Rule based orchestrator

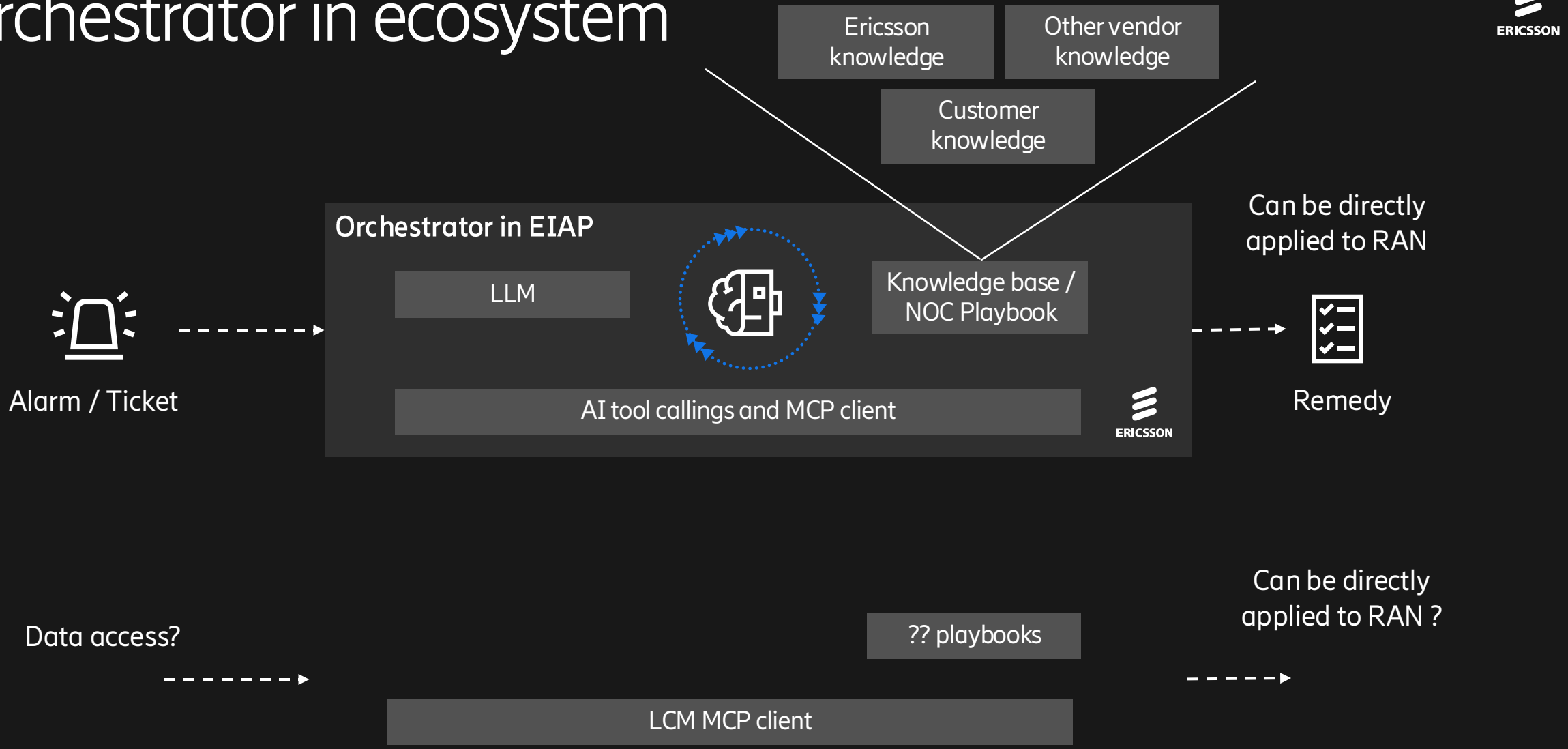


Basic AI orchestrator



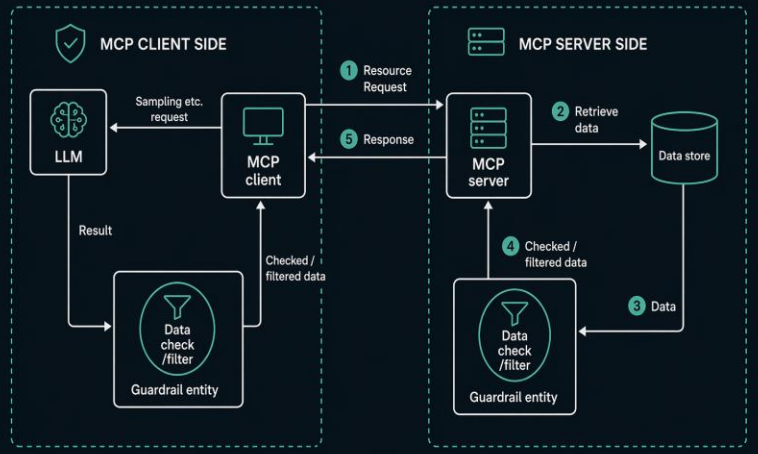
Advanced AI orchestrator

Orchestrator in ecosystem



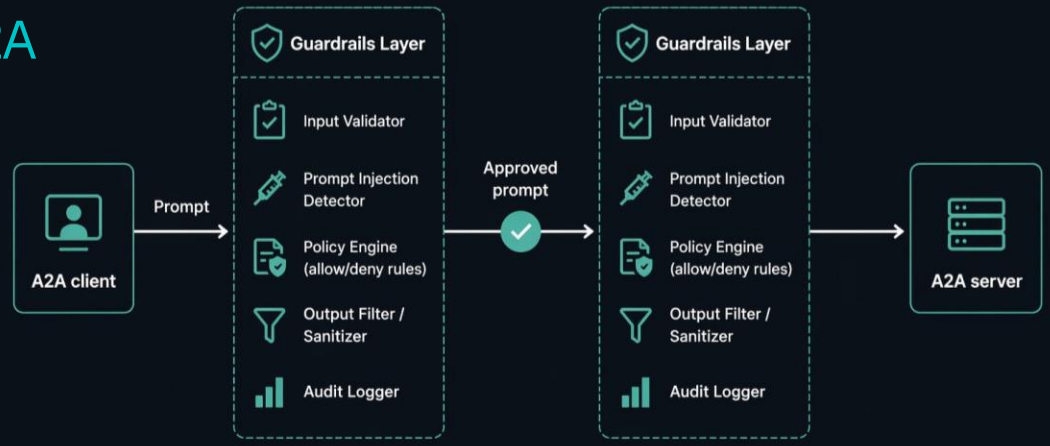
Security Considerations of Agentic AI

MCP



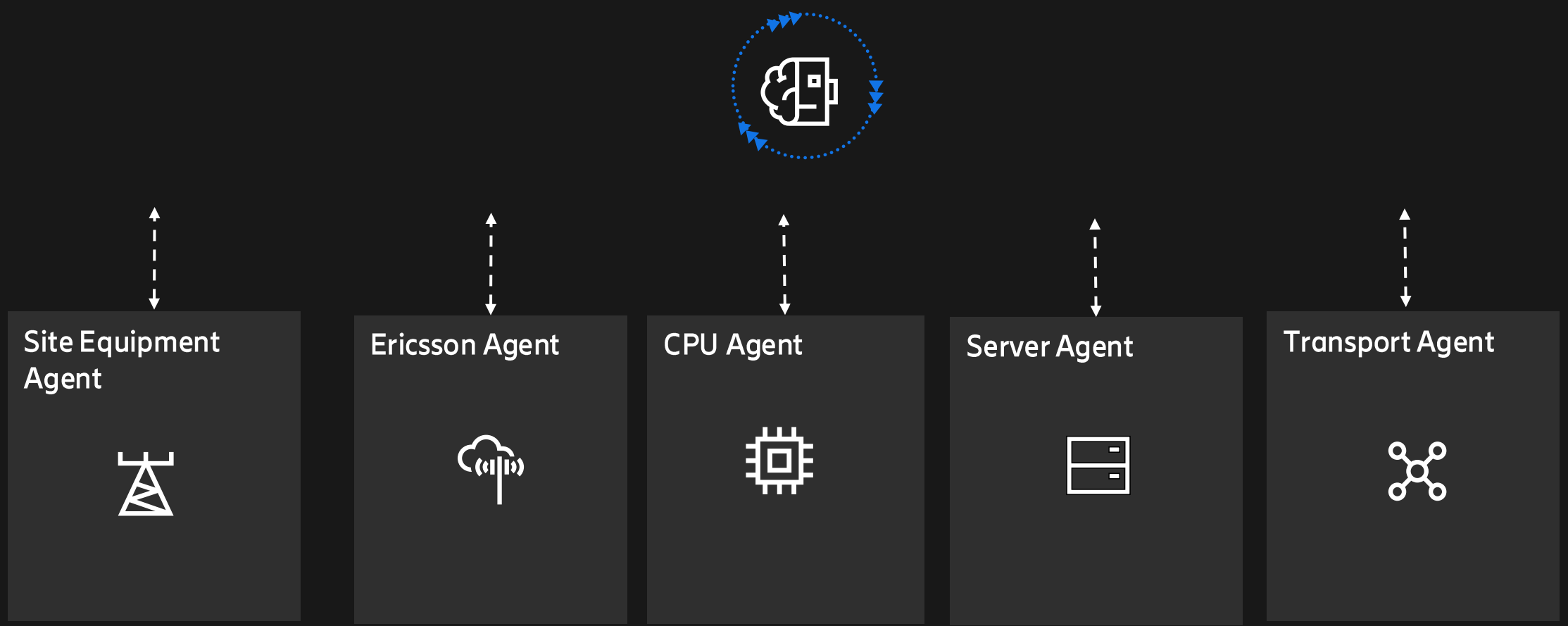
- **Model Context Protocol (MCP) has no built-in security:** organizations must design and enforce their own protection mechanisms
- **Security focuses on controlling access and protecting data:** verifying identities, limiting permissions using standard protocols
- **Guardrails are added externally:** to prevent exposure of sensitive information through filtering, redaction, or anonymization

A2A

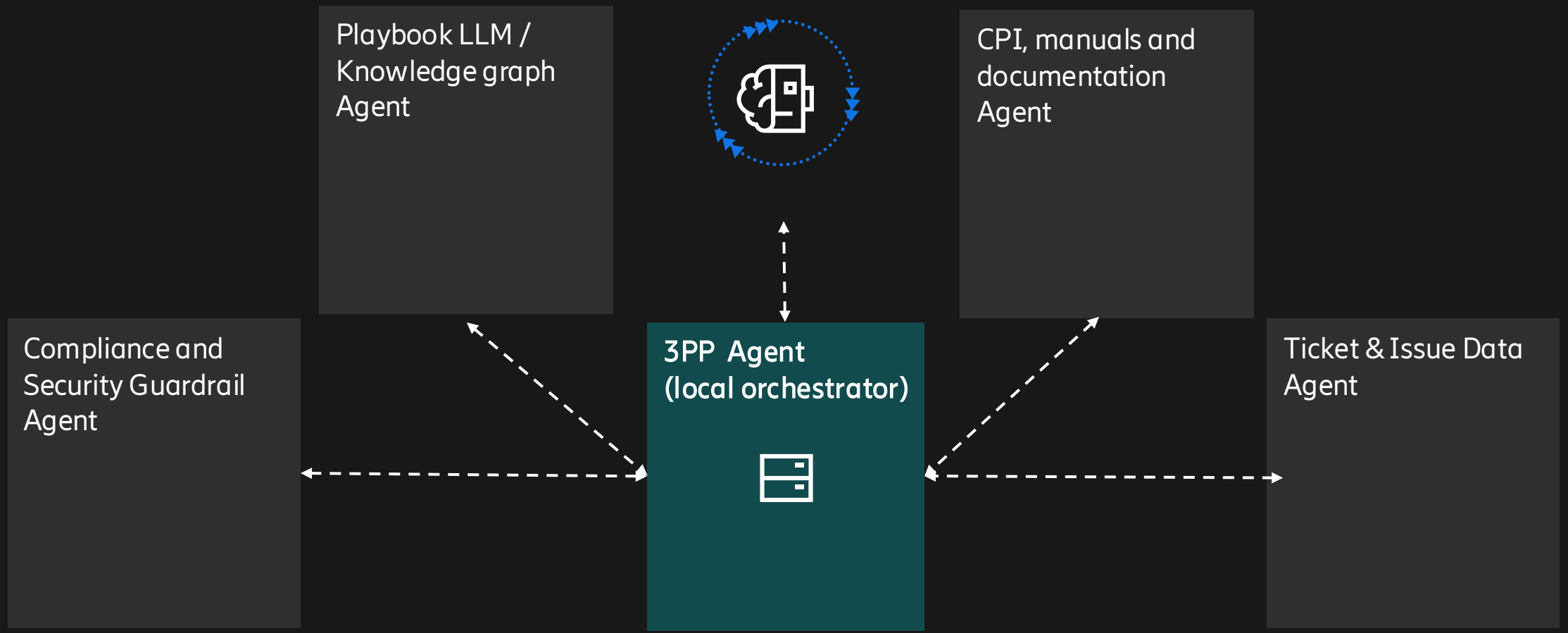


- **Agent-to-Agent (A2A) includes built-in security concepts:** defining how agents authenticate, exchange data, and enforce access control
- **Security focuses on trusted agent interactions:** verifying identity, enforcing permissions, and controlling task/data access
- **Operational safeguards are still required:** validation, monitoring, and governance ensure protection against misuse, manipulation, and data leakage

Multivendor Agents in Ecosystem



Multi-vendor Agents in Ecosystem



Agentic Troubleshooting PoC Learnings

Field Engineering	For the specific PTP / Synch use cases we explored in this PoC, we were able to identify the root cause within seconds and suggest resolutions that did not require FSE support
Orchestrator	Building the Orchestrator to include Ericsson knowledge base , clearly benefitted from methodology and training from Ericsson extensive customer and internal data
Agents	For efficient agents, all parties in the ecosystem needs to engage and create the needed components, however since the ownership of the Agents is decentralized, the development can be staggered and is not hinged on coordinated deliveries.
Knowledge Base	Including the knowledge base inside EIAP simplified the access to data and also the possibility to automatically remedy issues. It can also contain several workflows and continuously learn based on system feedback. This is also needing a larger training set to be production ready – the PoC focused on PTP sync issues in general
Ecosystem	An agentic solution built in the ecosystem allows for different timelines and effort and does not necessarily need to be synchronized between all parties.