

# Automate 5G Network Configurations with NVIDIA AI LLM Agents and Kinetica Accelerated Database

Geralyn Chong

March 18, 2025

## Contents

<b>1</b>	<b>5G-Slicing-lab Overview</b>	<b>1</b>
<b>2</b>	<b>Introduction to LLM-Agents</b>	<b>2</b>
<b>3</b>	<b>Agent Building Essentials</b>	<b>2</b>
3.1	ReAct Agent in Langgraph . . . . .	2
<b>4</b>	<b>5G Network Agent Overview</b>	<b>2</b>
4.1	5G-Network Architecture . . . . .	2

## 1 5G-Slicing-lab Overview

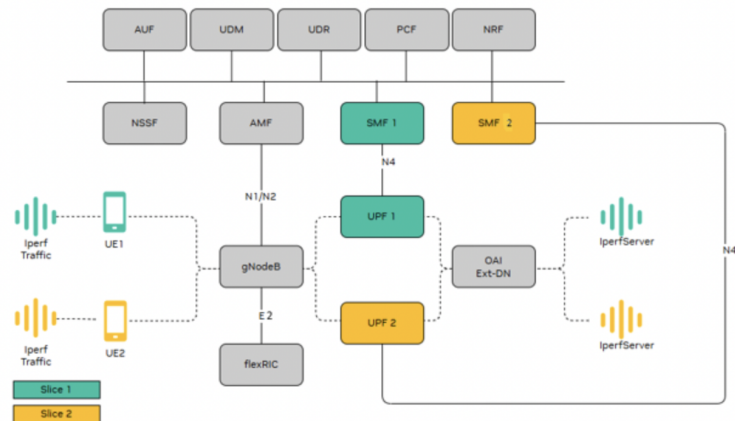
Kinetica Database powers sub-second analytics and vector search on large amounts of real-time data. PHINE.Tech is a 5G startup to enable and accelerate 5G use cases offering a virtual 5G lab like a lab-as-a-service.

Open Air Interface (OAI) is an open-source software stack.

### 5G Lab Scenario Setting

We have created a 5G Scenario using Open Air Interface Software Stack

The Open Air Interface (OAI) is an open-source software stack designed to implement the 3GPP wireless standards, facilitating research and development in mobile telecommunications technologies.



To simulate traffic that will go through the slices which share the gNodeB. How to configure with an LLM the allocation of slices in the gNodeB (100% capacity) depending on the traffic generated.

In the experiment, we include two User Equipments (UEs) across two different network slices, each dynamically adjusting their traffic rate between 30 Mbps and 120 Mbps every 100 milliseconds, with an AI gent dynamically managing slice allocations within the gNodeb. We will have a traffic generator with iPerf to the UE. Each time slot, the UEs are going to push different bit rates.

**Location?** This is a very basic setup why do we need an LLM? Idea: We will always have a perception / monitoring mechanism and an action / configuration mechanism in order to execute actions accordingly.

## 2 Introduction to LLM-Agents

LLM Agents autonomously performing tasks, make decisions, and interact with users or other systems.

- LLM/Agent Core - Central engine responsible for understanding and generating text. It defines agent's behavior and serves as the primary decision-making unit
- Planning module - determines the sequence of actions needed to accomplish a goal
- Memory Module
- Tools -

Orchestration Agent directs the more specialize agents that complete provided tasks broken down from the user / larger problem.

Agentic Frameworks - structured environments for building AI agents that can reason, plan, and execute tasks autonomously. Interactions between LLMs, memory, tools, and external environments.

**Langchain vs Langgraph**  
Comparing popular agentic frameworks

Feature	LangGraph	LangChain
Execution Model	Graph-based, parallel & stateful.	Sequential, tool-driven.
Best For	Multi-agent workflows, parallel tasks.	Single-agent apps, RAG, chatbots.
State Management	Stateful, retains intermediate results.	Stateless (can store memory).
Scalability	Handles complex workflows efficiently.	Designed for simpler use cases.
Tool Calling	Supports multiple tools in one step.	Calls tools sequentially.
Ease of Use	More structured, explicit workflows.	Easier for quick prototyping.

## 3 Agent Building Essentials

Tool Calling allows LLMs to invoke external functions, APIs, or utilities dynamically.

### 3.1 ReAct Agent in Langgraph

Agent Architecture that combines step-by-step reasoning with tool use. LangGraph provides a prebuilt function `create_react_agent()`

## 4 5G Network Agent Overview

### 4.1 5G-Network Architecture

We have two agents: Monitoring agent and the network configuration agent. The monitoring takes in the gNobE logs and parse the logs looking for errors. This error detection calls the network configuration agent which access the `get_pkt_loss_tool` and the `reconfigure_network_tool`. The `get_pkt_loss_tool` is connected to the Kinetica database retrieves the latest packet loss. Analyzing

the logs and determines which UE needs more bandwidth. Based on this, it assigns higher bandwidth to the selected UE. This triggers a call to the reconfigure network tool.