WSO2CON 2025

Building AI Applications in the Enterprise

Nadheesh Jihan
Technical Lead
WSO2

Our Mission Today!

I will be taking you on a journey:

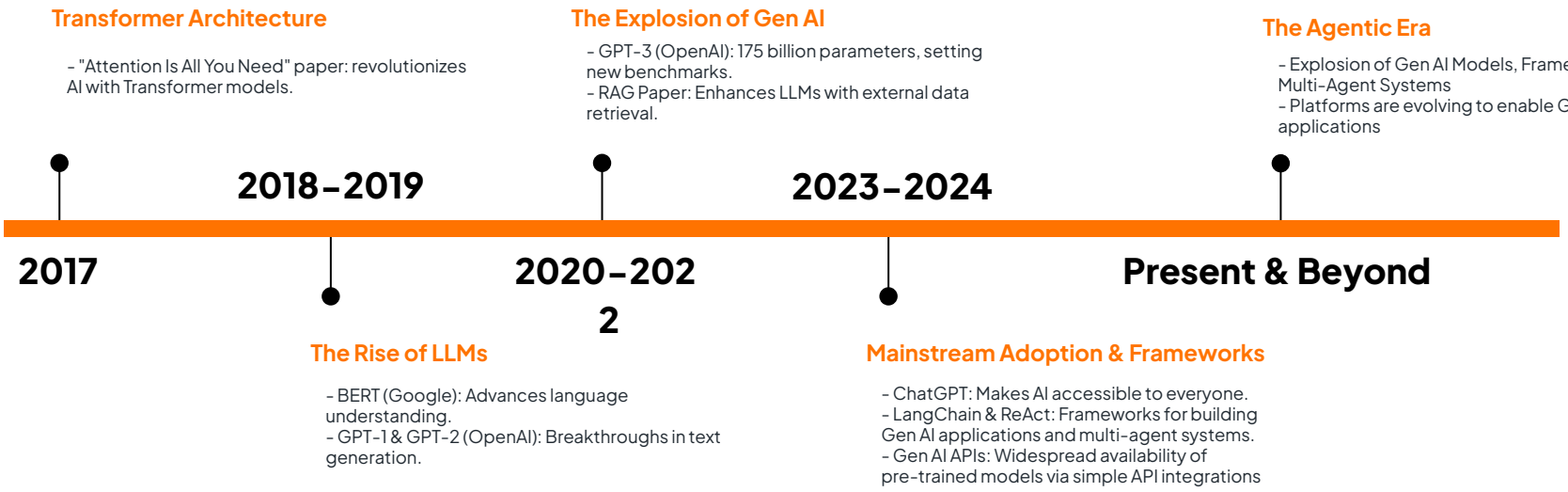
- **Practical Use-Case Evolution**: Guide a real-world use case to a scalable solution.
- **Gen AI Drawbacks**: Explore challenges and how to address them.
- **Key "Gen-AI Patterns"**: Learn three essential patterns for building Gen AI apps.
- **Hands-On Implementation**: Implement each pattern to see it in action.
- **Leveraging Integration Expertise**: Use integration skills to create successful Gen AI apps.
- **AI & Integrations**: Understand how AI and integration strategies intersect.

From Traditional AI to Generative AI

What is AI, Gen AI, and LLMs?

- Artificial Intelligence (AI)
  - Simulates human intelligence in machines to perform tasks like learning, problem-solving, and decision-making.
- Generative AI (Gen AI)
  - A type of AI that creates new, original content (text, images, music) by learning from existing data patterns.
- Large Language Models (LLMs)
  - A specific type of Gen AI that understands and generates human text.

Generative AI: A Significant Leap Forward!



Business Value: What's Changed?

- Building AI Applications was Previously:
  - Complex, time-consuming, and required specialized knowledge in AI, ML, Deep Learning, Data Science, etc.
- With Gen AI:
  - No need for deep AI/ML expertise.
  - Gen AI models are **pre-trained** and easily **accessible through APIs.**

Building AI applications is now all about coding and integrations!

Do You Want to Build an AI Application?

A Real-World AI Use Case

Enhancing WSO2Con with AI

We've integrated AI-driven features into the WSO2Con mobile app to improve your conference experience:

- **WSO2Con Assistant:** Get real-time answers to your questions about conference.
- **Session Advisor:** Receive personalized session recommendations based on your interests.
- **Attendee Connections:** Helps to connect with attendees who share similar interests.
- **Expert Finder:** Discover WSO2 O2Bar experts to discuss your specific technical needs.

A Real-World AI Use Case

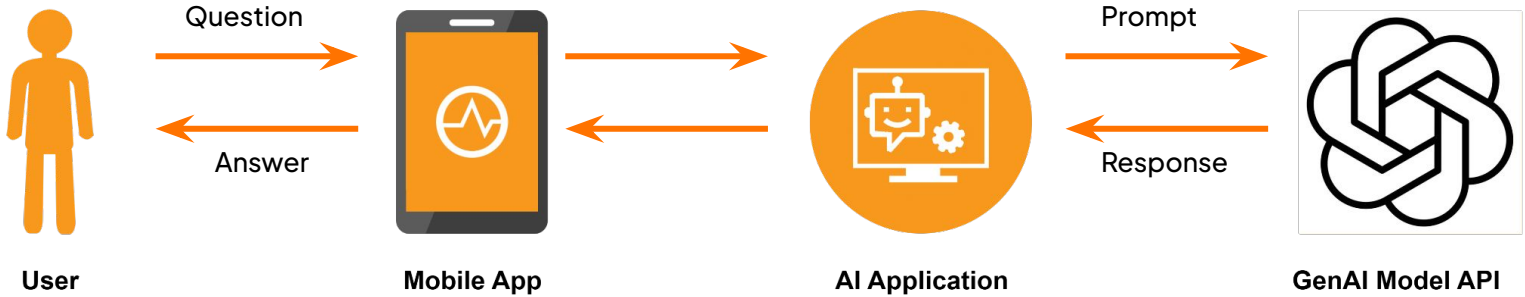
Enhancing WSO2Con with AI

We've integrated AI-driven features into the WSO2Con mobile app to improve your conference experience:

- **WSO2Con Assistant:** Get real-time answers to your questions about conference.
- **Session Advisor:** Receive personalized session recommendations based on your interests.
- **Attendee Connections:** Helps to connect with attendees who share similar interests.
- **Expert Finder:** Discover WSO2 O2Bar experts to discuss your specific technical needs.

Brainstorming!

- Building the WSO2Con Assistant
  - We have Generative AI models—so how hard can it be?
- A prompt is simply:
  - Instructions, context or questions that guides the model to generate a relevant response



Let’s Start: Building the WSO2Con AI Assistant

It Doesn’t Have All the Answers!

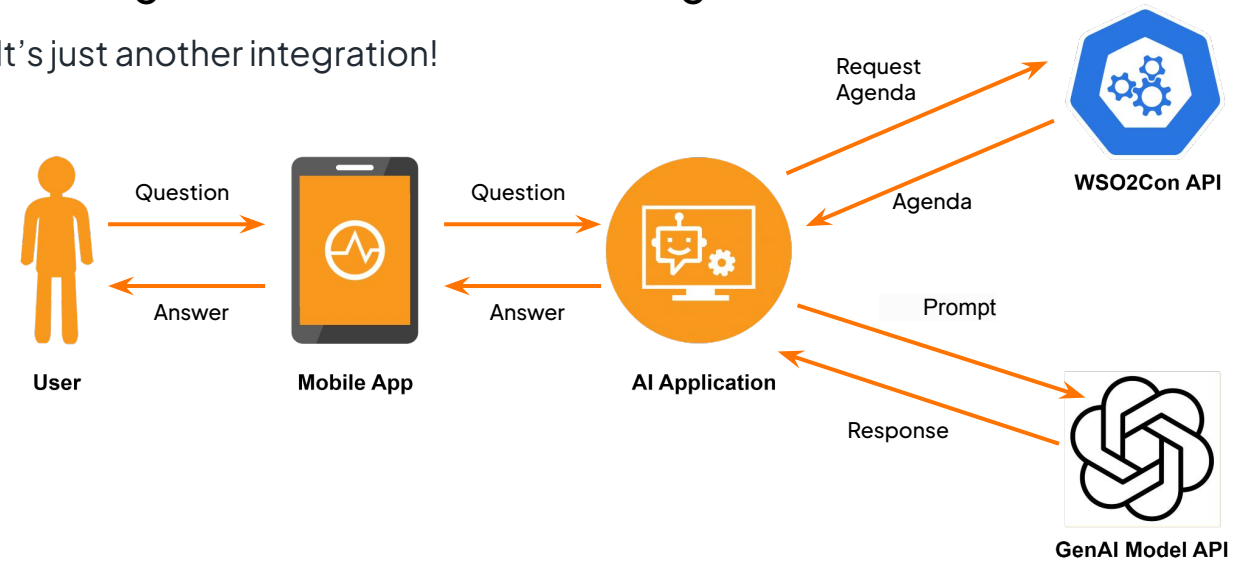
- Gen AI models have a knowledge cut-off.
  - They only "know" up to the point of their last training data.
- Key Warning:
  - They don’t provide facts—they generate responses based on patterns.
  - Hallucinations are a real risk.
- So how do we add knowledge to a Gen AI model?

Can We Train It to "Know" More?

- Fine-tuning can update Gen AI model with new knowledge.
- Require some expertise on how to:
  - Structure the data
  - Tune hyperparameters (epochs, learning rate, etc.)
- Takes time & isn't real-time
- Not ideal for WSO2Con Assistant → Conference agendas change frequently!

In-Context Learning – A Simpler Approach

- Grounding the Model with Facts
  - Instead of fine-tuning, we can inject real-time data into the prompt:
- Prompt = Instructions + Question + Live Data (Conference Agenda)
- How do we get the latest conference agenda?
  - It's just another integration!



WSO2Con AI Assistant with Knowledge Integration

Level Unlocked: AI Application Builder

- The assistant answers questions correctly using real-time agenda data.
  - But what if users ask about WSO2 products?
- We need more data sources—but there's a challenge!

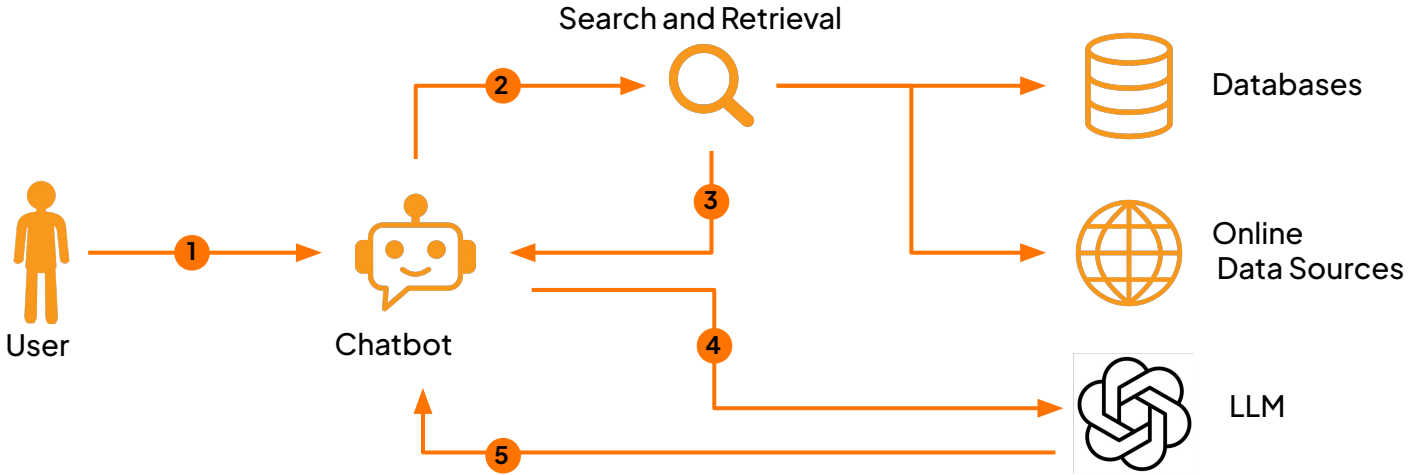
Dall-E generated

The Challenge – LLMs Have Token Limits

- LLMs can’t process unlimited data in one go!
  - For example: GPT-4o has a 128k token limit (input + output).
- That may seem like a lot, but…
  - 📉 Accuracy drops if we stuff too much data.
  - 🕑 Response time increases with longer prompts.
  - 💰 Costs go up as we send larger requests.
- We need a better approach!

Retrieval-Augmented Generation (RAG)

- Instead of stuffing all data into the prompt…
- Retrieve only the most relevant data on demand!
  - Efficient – Only fetch what's needed.
  - Accurate – Uses up-to-date, context-aware data.
  - Scalable – Works even with massive datasets.
- But how does it work? Let's break it down…



RAG Explained – Three Key Steps

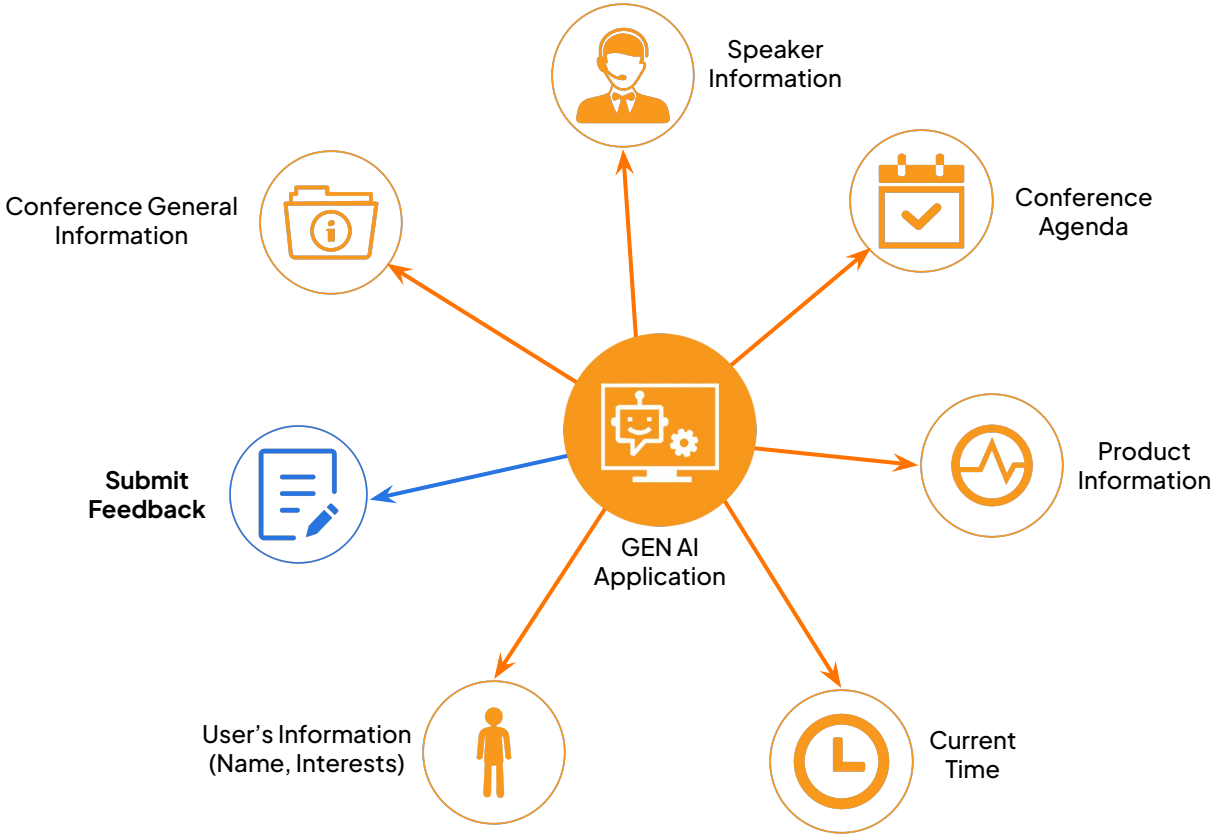
1) Data Ingestion (Happens Once)
   a) Chunk the information into smaller, meaningful sections.
   b) Convert each chunk into embeddings using an embedding model.
   c) Store embeddings in a vector database for efficient retrieval.
2) Data Retrieval (For Every Question)
   a) Convert the user's question into embeddings.
   b) Perform a similarity search in the vector database.
   c) Fetch the most relevant chunks.
3) Augmented Generation
   a) Include only relevant data in the prompt.
   b) Generate a fact-grounded answer using the LLM.

Let’s Supercharge Our AI Assistant with RAG!

Beyond Agenda & Docs – What's Next?

- Our WSO2Con Assistant can now answer agenda-related and product related questions.
- But what's missing?
  - Conference venue details
  - Information about speakers
  - Personalized recommendations based on attendees' interests
  - Ability to submit feedback
- Some of these require retrieving information, while others require storing new data.

Our Vision for the AI Assistant



How Do We Achieve This?

- At first glance, it seems like we just need more integrations.
- But think about it…
  - Should we load data from all sources for every question?
  - What if the user simply says, “Hi!” – do we need all that data?
  - How do we store user feedback dynamically?
- What if AI can do the heavy lifting?

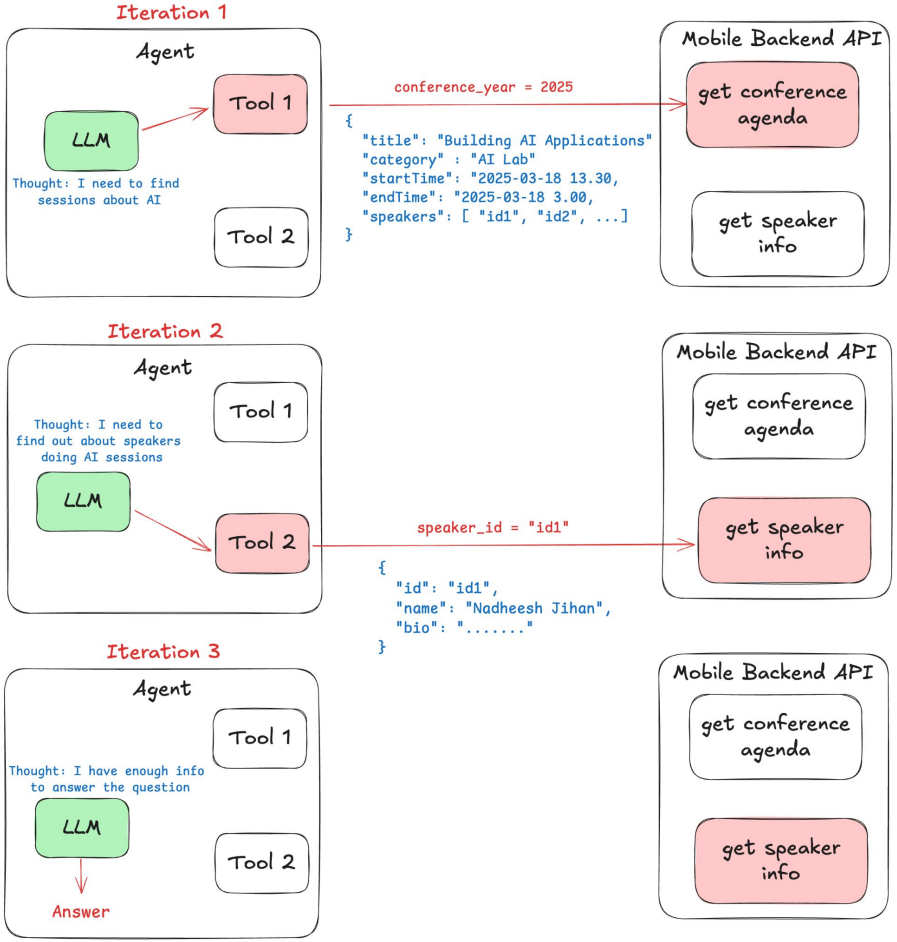
Let’s Make AI Do the Work!

- Instead of static workflows, we use a system that can reason and execute tools (APIs, functions, databases) dynamically.
  - Dynamic Data Retrieval – AI decides whether to query APIs, databases, or external systems based on user input.
  - Intelligent Action Execution – AI can invoke functions, trigger APIs, or update state as needed.
- These tools act as data sources or actions, chosen in real time by Gen AI.
- This is what we call a Gen AI Agent!

How Does a Gen AI Agent Work?

1) Agent loads a set of tools + a Gen AI model
2) Each tool has:
    a) A name, description, input schema, and execution method
3) The prompt instructs the LLM to reason and select the correct tool.
4) When a user asks a question:
    a) The agent decides which tool(s) to use
    b) Executes the tool dynamically
    c) Uses the output for the next reasoning step
5) This continues until:
    a) The task is fully completed
    b) The AI has enough information to respond to the user

AI Agent: Example



- Question: "**Who is speaking about AI at the conference?**"
- Iteration 1:
  - **Thought**: Find out sessions about AI
  - **Action:** Fetch the conference agenda
- Iteration 2:
  - **Thought**: Now figure out speaker names.
  - **Action**: Fetch speaker info
- Iteration 3:
  - **Thought**: I have all info needed
  - **Action**: Provide answer to the question.

WSO2Con Assistant with AI Agents

Great Job! You made it!

Our Journey So Far…

- We've explored three key patterns for building GenAI applications:
  - **Gen AI Integrations**: Connecting pre-trained models with enterprise systems via APIs.
  - **Retrieval-Augmented Generation (RAG):** Enhancing AI responses by dynamically retrieving relevant external data.
  - **GenAI Agents**: Autonomous systems that can make decisions, and execute tools dynamically based on the given task.
- We built a GenAI-powered application in just a few minutes—seems simple, right?
- But despite the excitement, very few GenAI applications actually succeed in production. Why?

A Common Pitfall!

- Many organizations are investing heavily in GenAI applications, but only a small fraction make it to production.
- What's going wrong?
  - Some see it purely as an AI problem (model tuning, prompting, etc.).
  - Others see it solely as an integration problem (APIs, data pipelines).
  - The reality? A successful GenAI application requires both AI and integration strategies!
  - And no—we're not talking about traditional AI expertise alone.

What Makes a Successful GenAI Application?

- Beyond just solving a valid problem, a production-ready GenAI application must:
  - Deliver accurate, consistent responses
  - Maintain acceptable latency for real-time interactions
  - Scale efficiently under varying loads
  - Securely access data and tools
  - Minimize misuse and unintended behavior
  - Provide explainability and visibility into its decisions

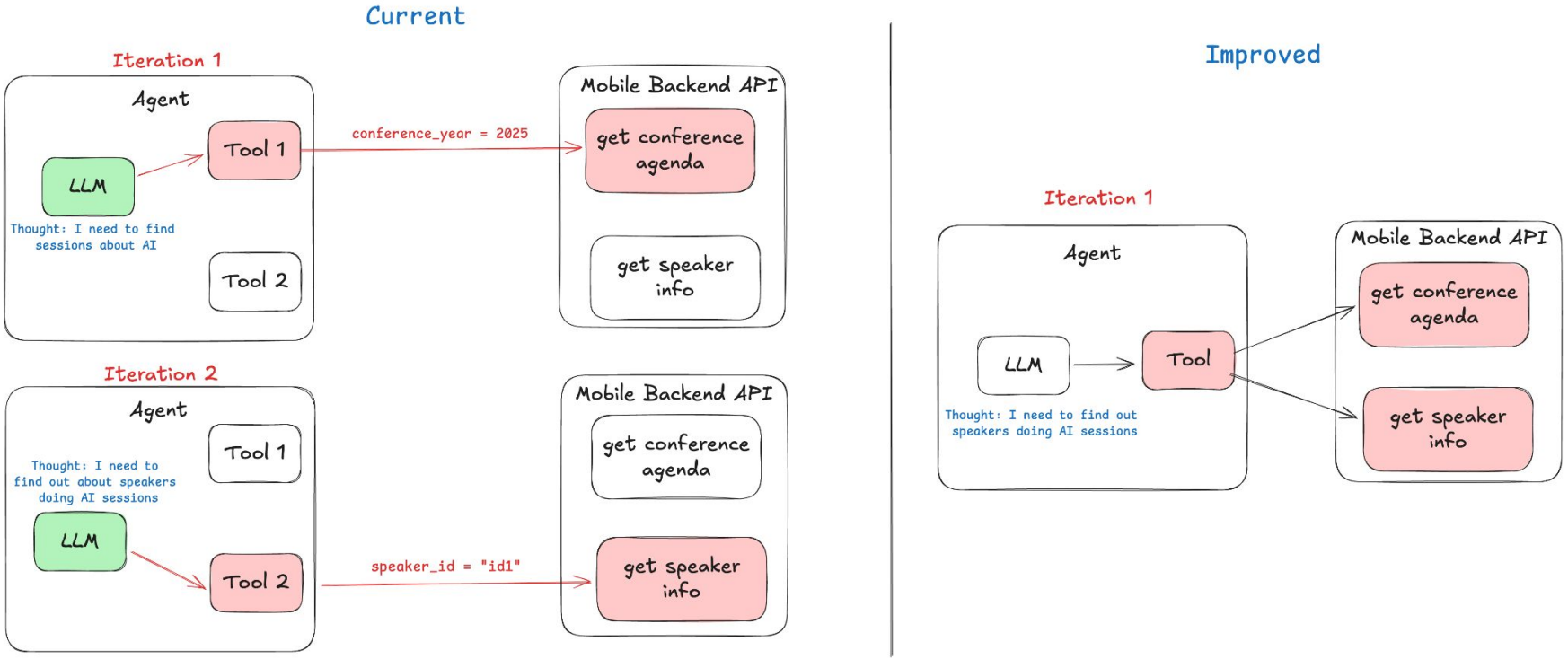
Improving WSO2Con Assistant

- What AI strategies can improve accuracy?
  - Prompt tuning (a.k.a. prompt engineering)
  - Hyperparameter optimization
  - Fine-tuning LLMs
- But is that enough? To take it to the next level, can we optimize the integration!

Taking A Step Back

- Let’s revisit the previous scenario:
  - Question: "Who is speaking about AI at the conference?"
- For an accurate response, the AI Agent must:
  - Query conference_agenda to identify AI-related sessions.
  - Query speaker_info to retrieve speaker details.
  - Map sessions to speakers using speaker IDs.
- Potential Risks:
  - Incorrect decision-making could result in returning only speaker IDs.
  - Faulty mappings could lead to incorrect speaker names.
- Key Insight: Recognizing these risks requires an AI-aware perspective

To Solve This, We Need Better Integrations!



Integration Expertise to the Rescue!

- Our modifications bring key benefits:
  - **Improved Accuracy:** Eliminates the risk of returning speaker IDs instead of names and prevents errors from incorrect mappings.
  - **Lower Latency:** Reduces unnecessary LLM calls and enables parallel execution of API requests.
  - **Cost Efficiency:** Optimizes token usage and minimizes redundant LLM calls.
- Why AI + Integration Expertise Matters
  - Traditional integration best practices may not always optimize AI-powered applications.
  - APIs are typically designed through domain-driven decomposition, but AI Agents require a different abstraction.
  - To build effective AI applications, we must think beyond traditional integrations and consider AI-driven optimizations.

How Can an AI Perspective Help?

- At the start, we explored four key AI-driven experiences:

  WSO2Con Assistant | Session Advisor | Attendee Connections | Expert Finder

- Should we apply the AI Agent pattern to all of them?
- As an integration developer, you might say yes
  - More automation is always better!
- But from an AI-first perspective, the decision won't be that straightforward.
  - Session Advisor: Gen AI integration
  - Expert Finder: Gen AI integration
  - Attendee Connections: Retrieval Augmentation Generation (RAG) + Gen AI Integration

After the Break!

- Rosen will share his insights on AI development and its impact on the industry.
- Afterwards, I'll walk you through the architecture behind the WSO2Con AI features and reveal how we moved from local development to production in just a few hours.
- Finally, I'll highlight the role of the AI Gateway in securing and optimizing AI interactions, using the WSO2Con use cases as an example.

Question Time!


Thank you!
WSO2CON2025