Ballerina + AI: Exploring AI capabilities in Ballerina

08/2025

Hello!

Sasindu Alahakoon

sasindu@wso2.com | Senior Software Engineer **|** **@ballerinalang** **|** **WSO2**

About this Session

Coming Up

- What is Ballerina
- What is AI
- Core concepts of AI
- AI for Ballerina
- Ballerina for AI
- Demonstration

What is Ballerina

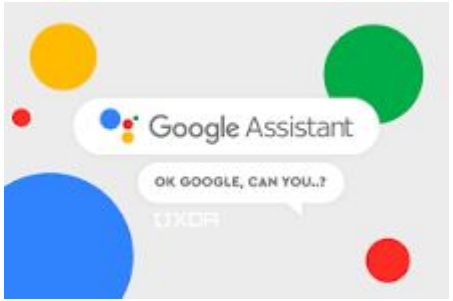
What is Ballerina

- Open source
- Cloud-native programming language
- Optimized for integration
- Rich ecosystem of network protocols, data formats, and connectors

What is AI

What is AI?

- Focused on creating machines that can simulate human intelligence
- Examples
  - Self-driving cars
  - Virtual assistants like Google Assistant or Siri



Core Concepts of AI

Large Language Model (LLM)

- A subset of AI
- Specializes in understanding and generating human language
- Capabilities
  - Answering questions
  - Summarizing documents
  - Generating code.
- Examples
  - Google’s Gemini
  - OpenAI’s GPT series (e.g., ChatGPT)

Roadblocks of using standard LLMs in real world

- The Knowledge Cut-off Problem
  - Only knows what it was trained on (frozen in time)
- The Hallucination Problem
  - Make up facts when they don't know the answer
- Just Talking
  - Can't take action

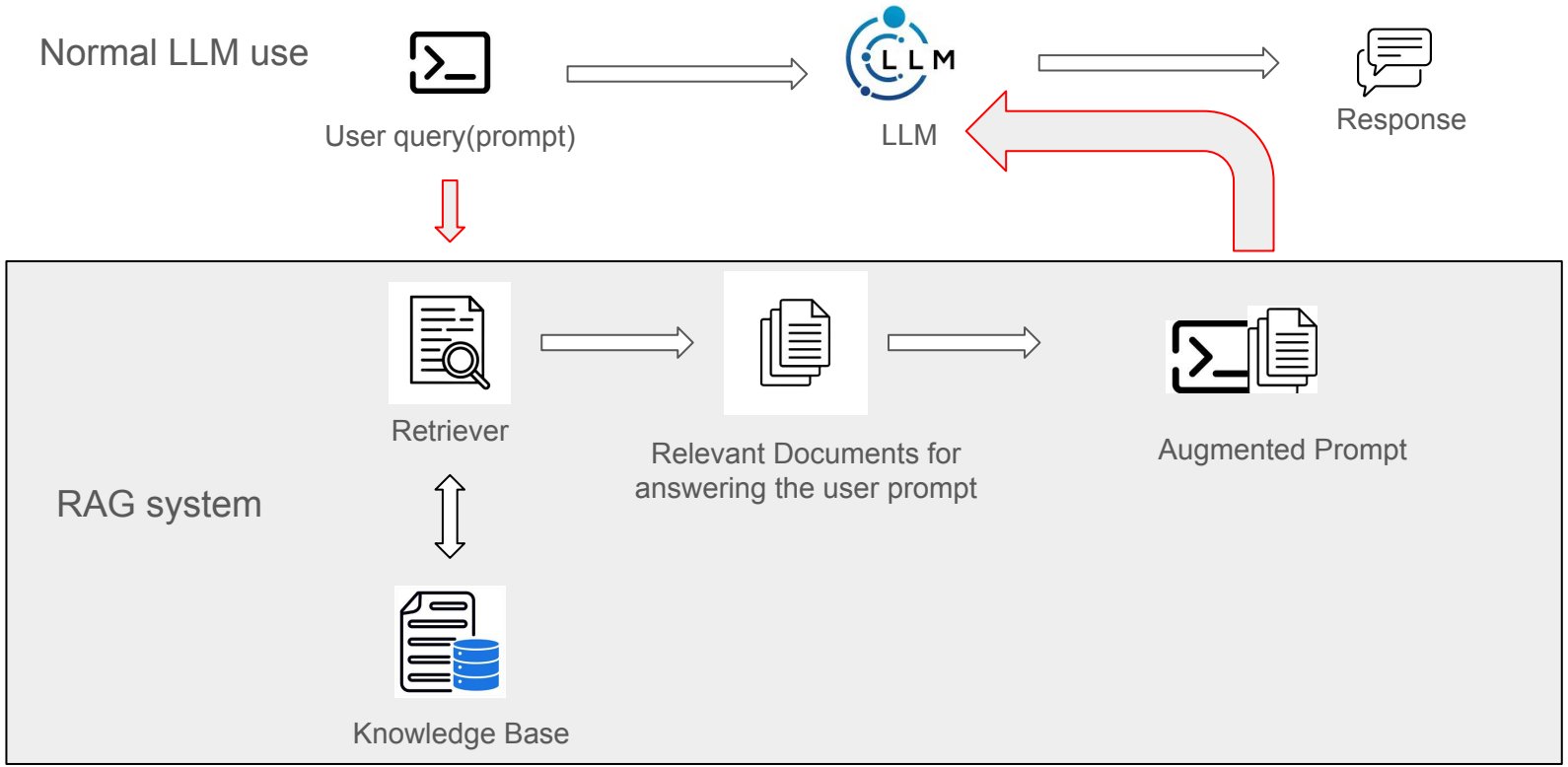
AI Applications

- Retrieval-Augmented Generation (RAG)
  - Up-to-date data
  - Factual knowledge
  - Examples
    - Give the knowledge to LLM about company policies, websites, internal documents
- Agents and Tools
  - Ability to perform actions
  - Example use cases
    - Booking a meeting, sending an email, updating a customer record
- MCP (Model-Context-protocol)
  - Standardizes interactions with external systems

Retrieval-Augmented Generation (RAG)

- Makes a LLM smarter by connecting it to a factual knowledge base
- Why RAG?
  - LLM knows a lot about the internet, But It knows nothing about company specific informations
  - RAG allows the LLM to securely access and use this private data
  - Eliminate Hallucinations
    - Sometimes the answer is **not just wrong**—it's **dangerous**.
    - RAG forces to answer on specific facts retrieved from a trusted knowledge base
  - Avoid the "Generic Knowledge" Problem
- RAG turns a generic AI into a personalized expert

RAG architecture overview



Prompt vs Augmented Prompt

- Prompt

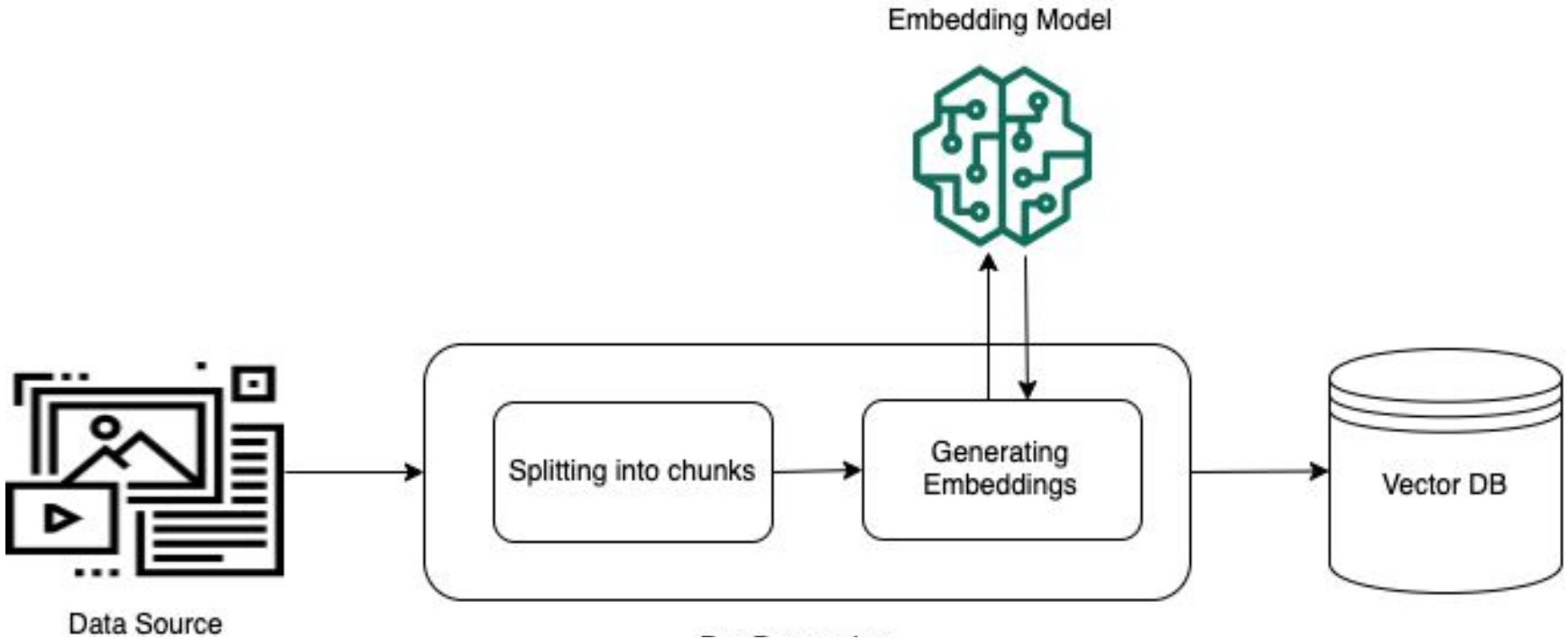
  "I have 5 unused annual leave days for 2025. Can I carry them forward to 2026 at Innovate Lanka?"

- Augmented Prompt

  CONTEXT: *"Section 7.4: Unused Annual Leave. Employees are entitled to carry forward a maximum of seven (7) unused annual leave days into the next calendar year. To be eligible, the employee must have completed a minimum of one full year of continuous service... All carried-forward leave must be utilized within the first quarter (i.e., by March 31st) of the subsequent year..."*

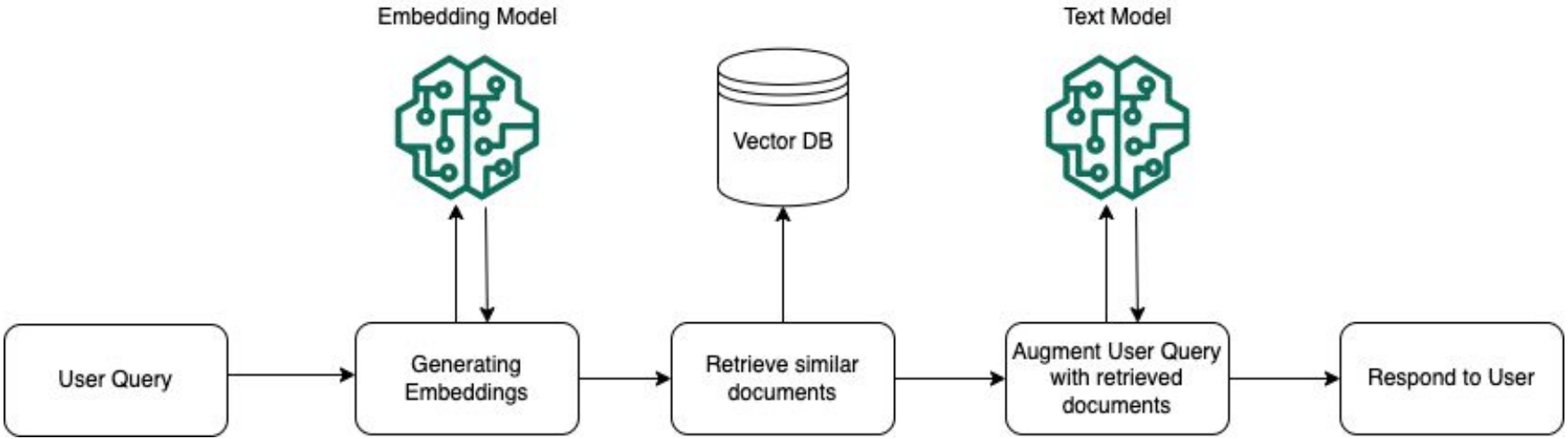
  USER'S QUESTION: *"I have 5 unused annual leave days for 2025. Can I carry them forward to 2026 at Innovate Lanka?"*

Rag Ingestion (Building the Knowledge Base for RAG)



Reference: https://docs.aws.amazon.com/bedrock/latest/userguide/kb-how-it-works.html

RAG Query



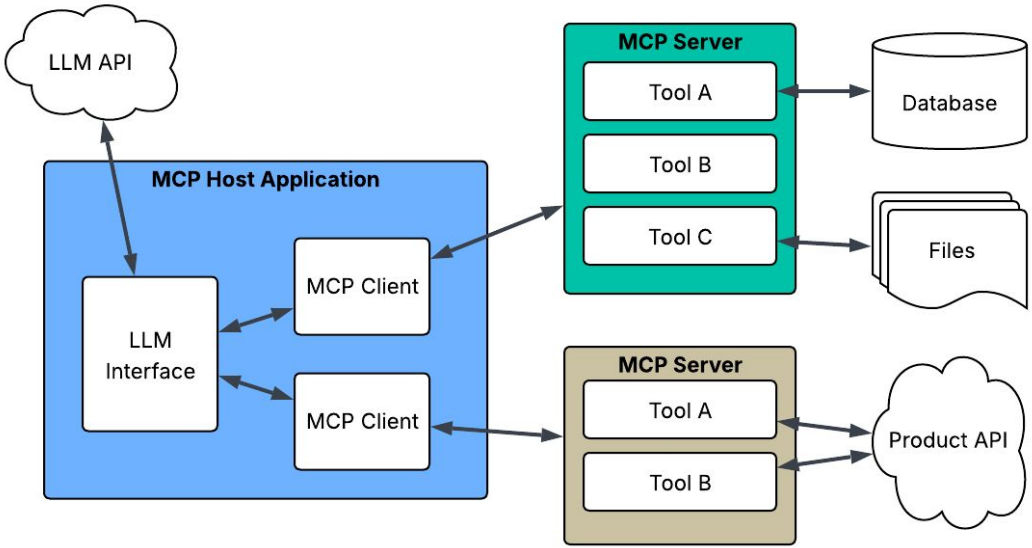
Reference: https://docs.aws.amazon.com/bedrock/latest/userguide/kb-how-it-works.html

AI Agents and tools

- What is a tool?
  - A specific function or capability you give to an LLM
    - `send_email(recipient, subject, body)` tool that can access your email client.
    - Simple calculator tool for math problems
- What is an agent?
  - LLM prompted to think strategically
  - Agent has brain, memory, and tools to interact with the world
  - The brain(LLM), that uses the tools and acting as the core reasoning engine
  - Memory to maintain context and learn from interactions

Model context protocol (MCP)

- Standardizes How AI applications interact with external systems
- Based on client-server architecture

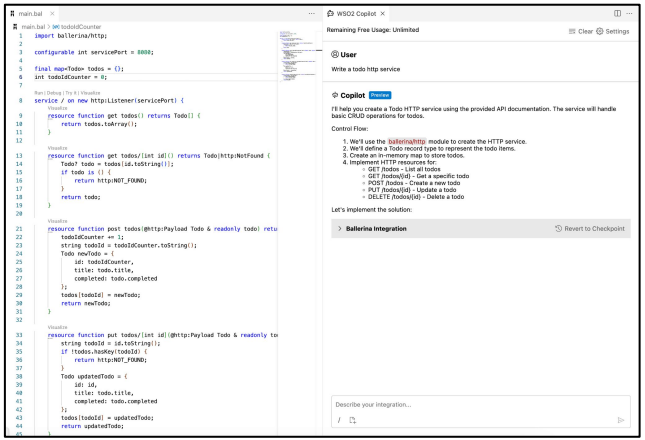


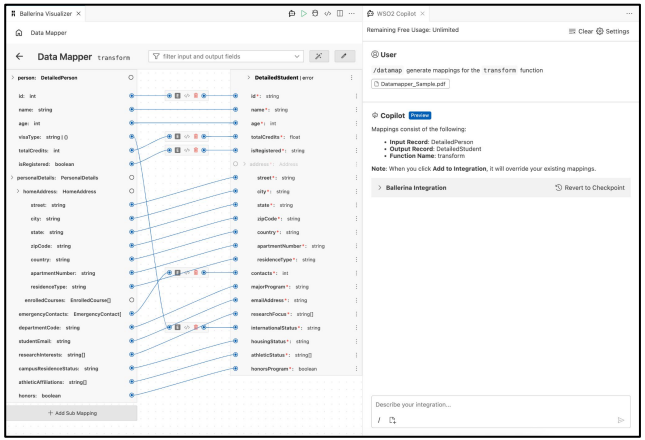
Reference: https://securityboulevard.com/2025/04/frequently-asked-questions-about-model-context-protocol-mcp-and-integrating-with-ai-for-agentic-applications/

AI for Ballerina

Ballerina Copilot

- Code generation using Natural Language
- AI data mapping
- Ask about Ballerina concepts, libraries and capabilities
- Test generation using Natural Language

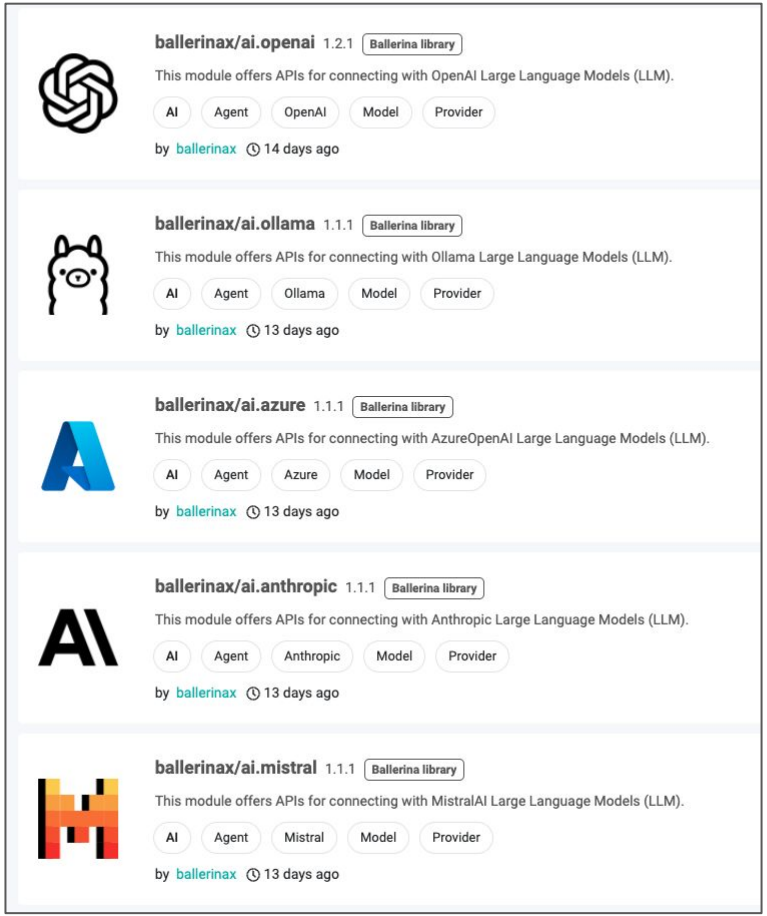




Ballerina for AI

Ballerina AI Connectors

- WSO2 Default Model Provider
  - Test all the Ballerina features entirely FREE
- Support various different model providers
  - OpenAI
  - Azure OpenAI
  - Anthropic
  - Deepseek
  - Mistral
  - Ollama



Direct LLM calls with Ballerina

- Support through `generate` API in model providers
- Designed for simple, stateless LLM calls where conversational history is not required
- Automatically structures the LLM response to your desired, type-safe format (e.g., JSON, Ballerina records, integer).
- Supports multimodal inputs for LLM prompts

Direct LLM Calls with WSO2 Default Model Provider

```ballerina
import ballerina/ai;

final readonly & string[] categories = ["Gardening", "Sports", "Health", "Technology", "Travel"];

Document this | Visualize
public type Blog record {|
    string title;
    string content;
|};

Visualize
type Review record {|
    string? suggestedCategory;
    int rating;
|};

Visualize
function reviewBlog(Blog blog) returns Review|error {
    ai:Wso2ModelProvider model = check ai:getDefaultModelProvider();

    Review review = check model->generate(`You are an expert content reviewer for a blog site that
        categorizes posts under the following categories: ${categories}

        Your tasks are:
        1. Suggest a suitable category for the blog from exactly the specified categories.
        If there is no match, use null.

        2. Rate the blog post on a scale of 1 to 10 based on the following criteria

        Here is the blog post content:

        Title: ${blog.title}
        Content: ${blog.content}`);

    return review;
}
```

Direct LLM Calls with OpenAI

```ballerina
import ballerinax/ai.openai;

configurable string apiKey = ?;
final readonly & string[] categories = ["Gardening", "Sports", "Health", "Technology", "Travel"];

Document this | Visualize
public type Blog record {|
    string title;
    string content;
|};

Visualize
type Review record {|
    string? suggestedCategory;
    int rating;
|};

Visualize
function reviewBlog(Blog blog) returns Review|error {
    openai:ModelProvider model = check new (apiKey, openai:GPT_4O_MINI);

    Review review = check model->generate(`You are an expert content reviewer for a blog site that
        categorizes posts under the following categories: ${categories}

        Your tasks are:
        1. Suggest a suitable category for the blog from exactly the specified categories.
        If there is no match, use null.

        2. Rate the blog post on a scale of 1 to 10 based on the following criteria

        Here is the blog post content:

        Title: ${blog.title}
        Content: ${blog.content}`);

    return review;
}
```

Direct LLM calls with Images in the prompt

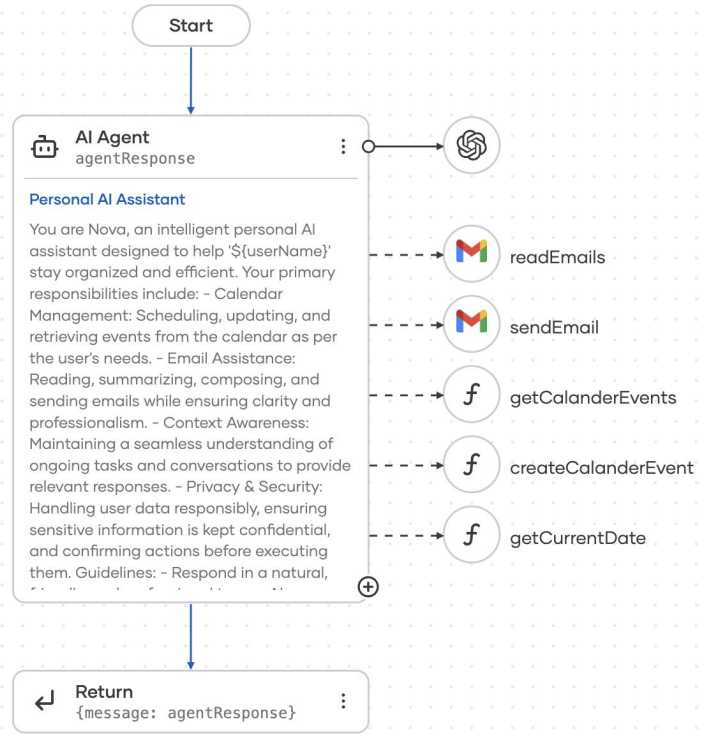
```
Run | Debug | Visualize | Try it
# Claims Processing API Service
#
# This service provides endpoints for processing insurance claims with AI-powered analysis.
service /insuarance on new http:Listener(8080) {
    Visualize
    # Process a new insurance claim submission
    #
    # This resource function accepts multipart form data containing a claim description
    # and an associated image. It uses AI model providers to analyze the content and
    # generate an intelligent summary of the claim.
    resource function post claims(http:Request request) returns ClaimResponse|error {
        mime:Entity[] bodyParts = check request.getBodyParts();

        string description = check bodyParts[0].getText();
        byte[] claimImage = check bodyParts[1].getByteArray();
        ai:ImageDocument claimImageDocument = {
            content: claimImage
        };

        ai:Wso2ModelProvider modelProvider = check ai:getDefaultModelProvider();
        string summary = check modelProvider->generate(
            `Please summarize the following claim
                - Description: ${description}
                - Image of the claim: ${claimImageDocument}`);
        return {
            submissionStatus: SUCCESS,
            summary
        };
    }
}
```

Building RAG applications with Ballerina

- Ballerina simplifies the development of complex RAG applications with its native support for AI
  - Automated Document Chunking
  - Effortless Vector Store Creation
  - Seamless Data Ingestion
  - Intelligent Context Retrieval

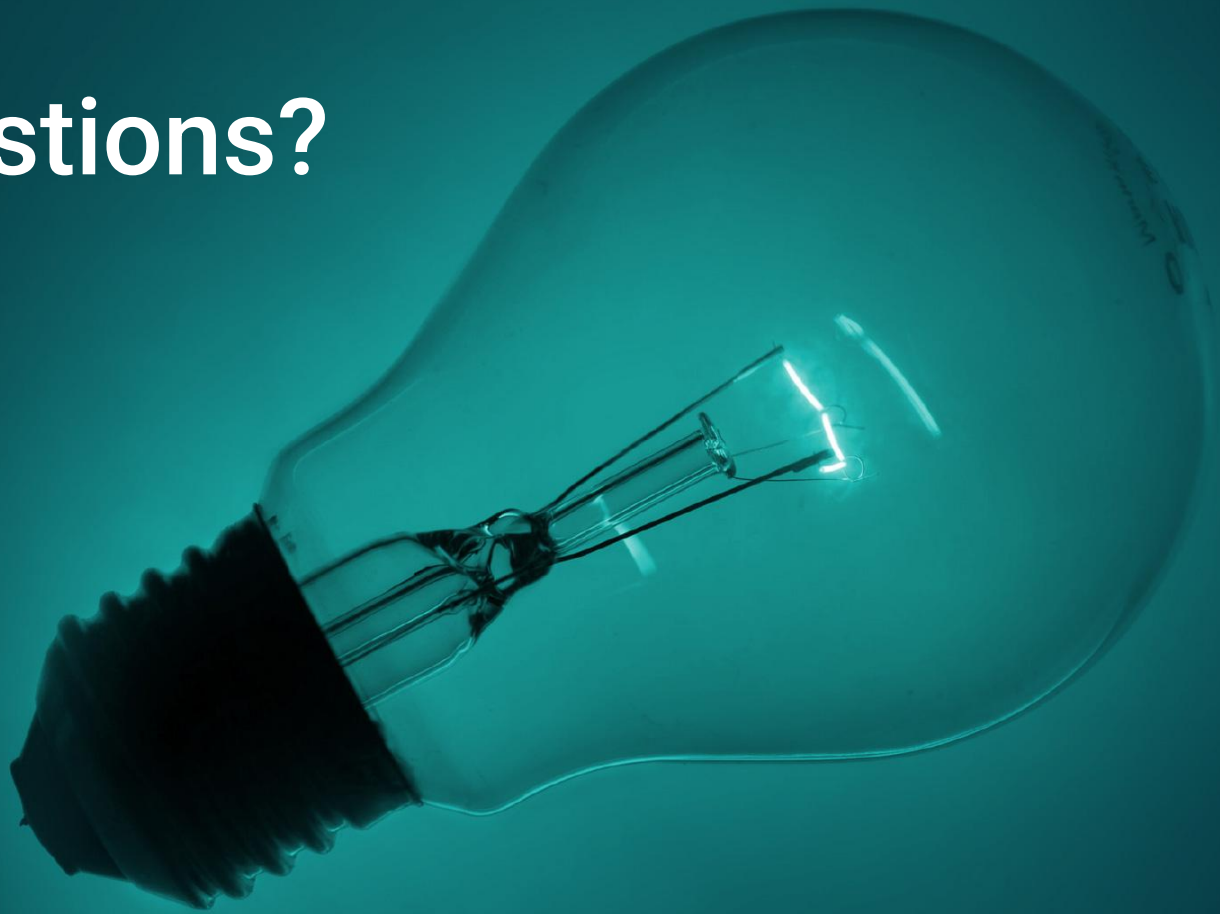
Building AI Agents with Ballerina

- Extendable via Ballerina's built-in connectors
- Chat Agents
  - Exposed through HTTP endpoints as REST APIs
  - Designed to interact with users or external systems
  - Ideal for chatbot-like experience
- Inline Agents
  - Embedded within service logic
  - Invoked programmatically as part of a backend workflow
  - Ideal for automation



Demonstration

Questions?



Learning resources

- Ballerina documentation and tutorials
  - Learn Guide
    - ballerina.io/learn/
  - Ballerina by example
    - ballerina.io/learn/by-example
  - Ballerina for AI
    - https://ballerina.io/use-cases/ai/
  - YT Training Series

Community Channels

https://github.com/ballerina-platform/

https://stackoverflow.com/questions/tagged/ballerina

https://discord.com/invite/ballerinalang

https://twitter.com/ballerinalang

Ballerina student program

- Ballerina student engagement program
  https://ballerina.io/community/student-program/
- Ballerina ambassador program
  https://ballerina.io/community/ambassadors/

Thank you!