

20TH ANNIVERSARY EDITION

WSO2CONASIA

— PLATFORMLESS MODERNIZATION

Augmenting Integration with Autonomous AI Agents



Shafreen Anfar

Architect/AI Lead of Integration department

WSO2



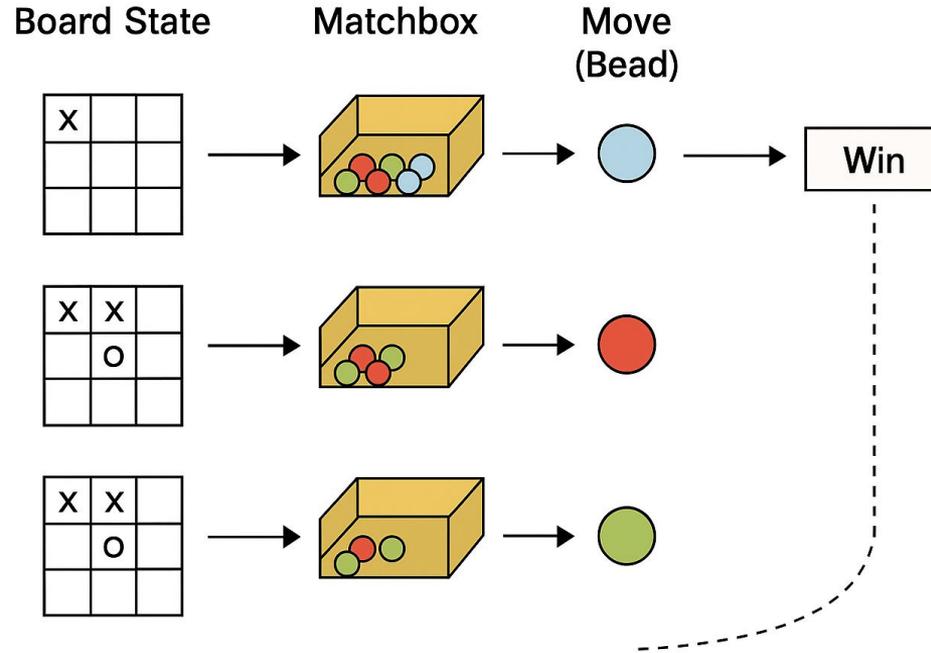
I help develop tools that enable integration developers to build amazing things!



Matchboxes



Matchboxes





I help integration developers build amazing things by building beautifully designed, simple and easy to use tools.



Agenda

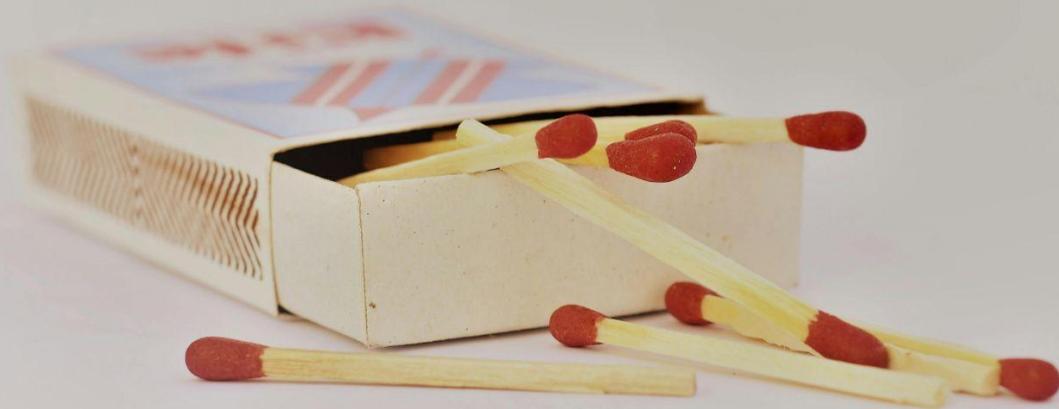
- Augmenting Integration with Autonomous Agents (20 mins)
 - Intelligence
 - Agents
 - Integration

AI's biggest challenge isn't intelligence — it's integration.



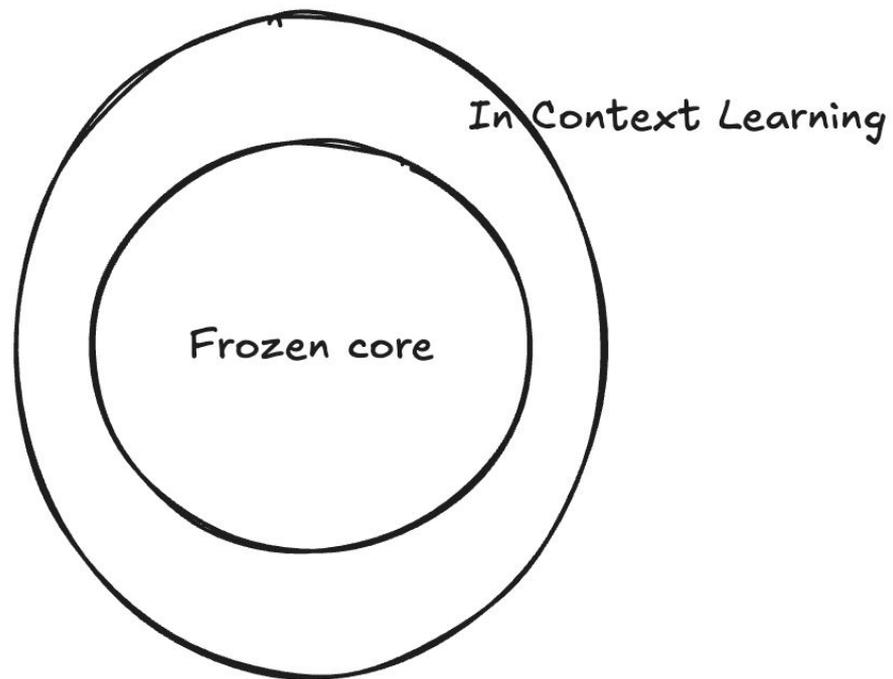
Intelligence

AI began with matchboxes.

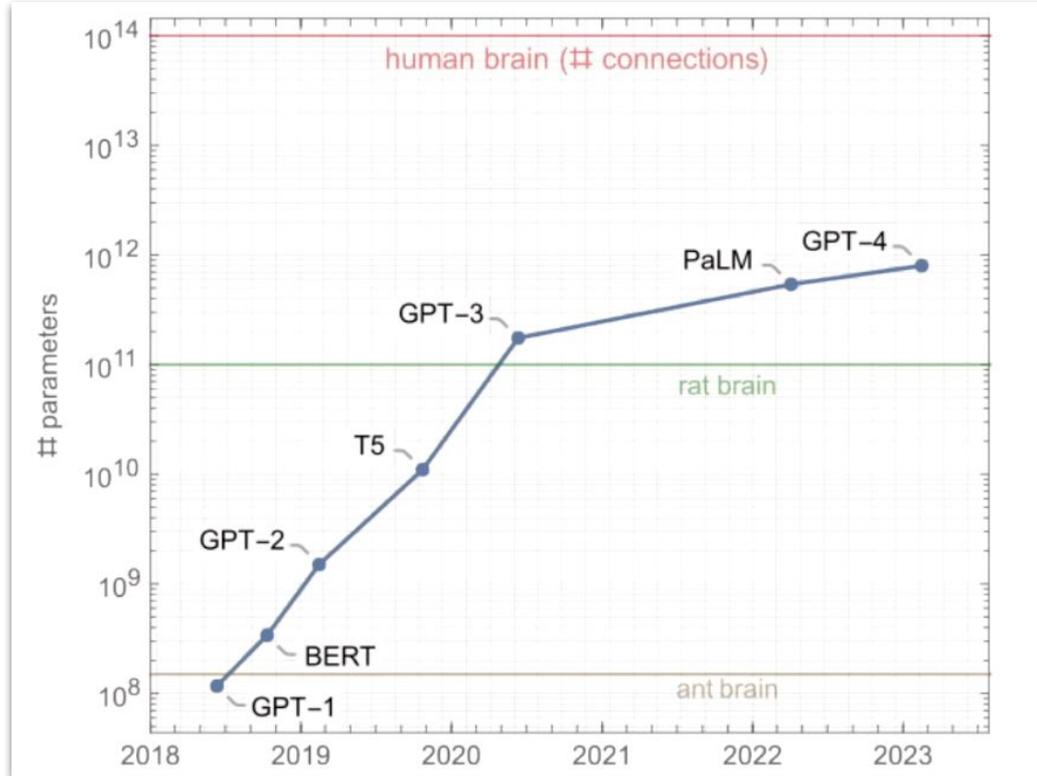


General Purpose Language Model

General Purpose Language Model



General Purpose Language Model



Source: https://www.youtube.com/watch?v=_6R7Ym6Vy_I

They are just an API call away...

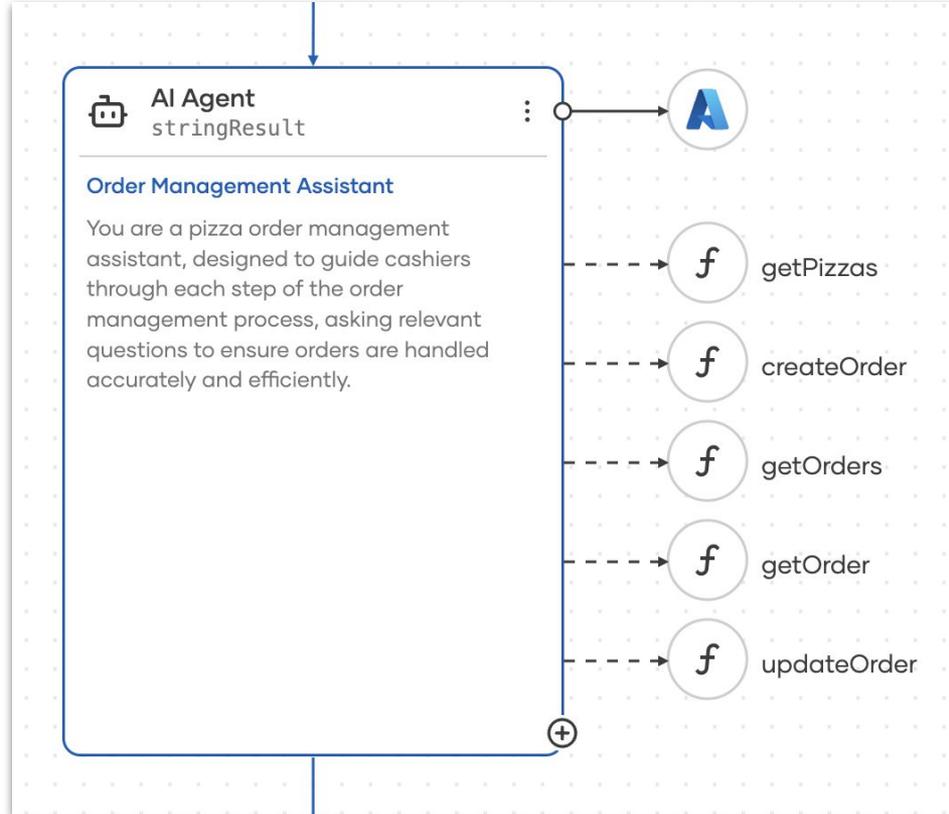


Agents

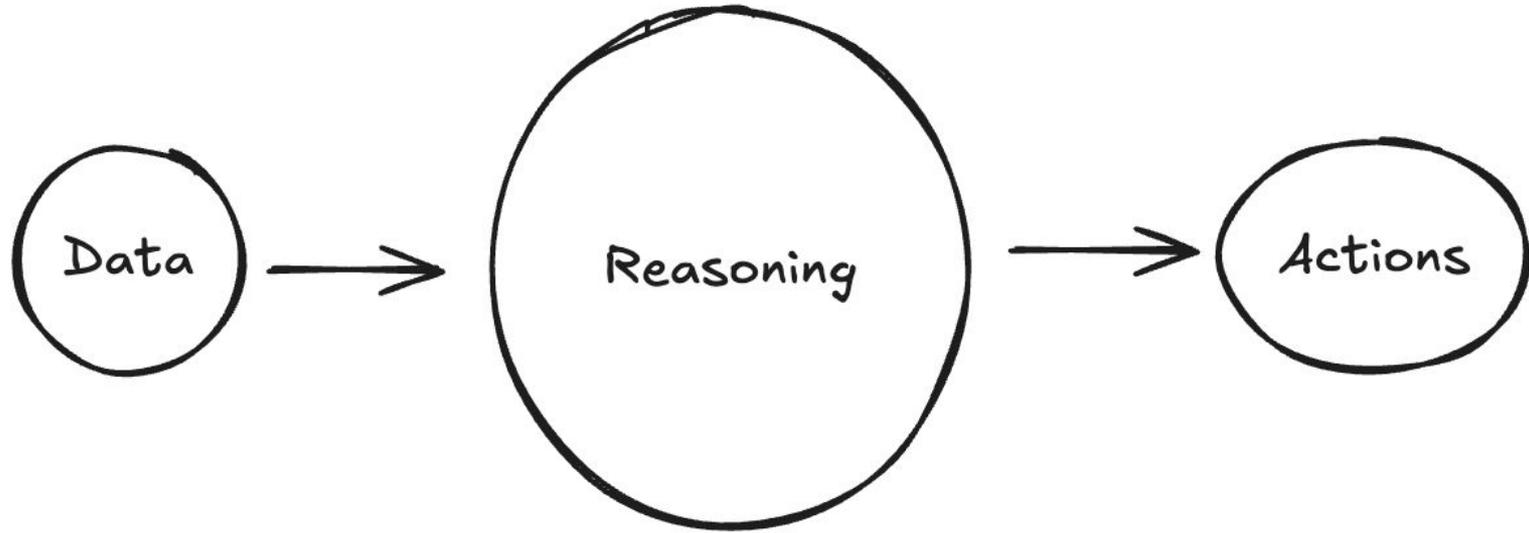
Unlocks the door to a new world



Agent Building Blocks



Agent behaviour



Intelligence → Data → Integration

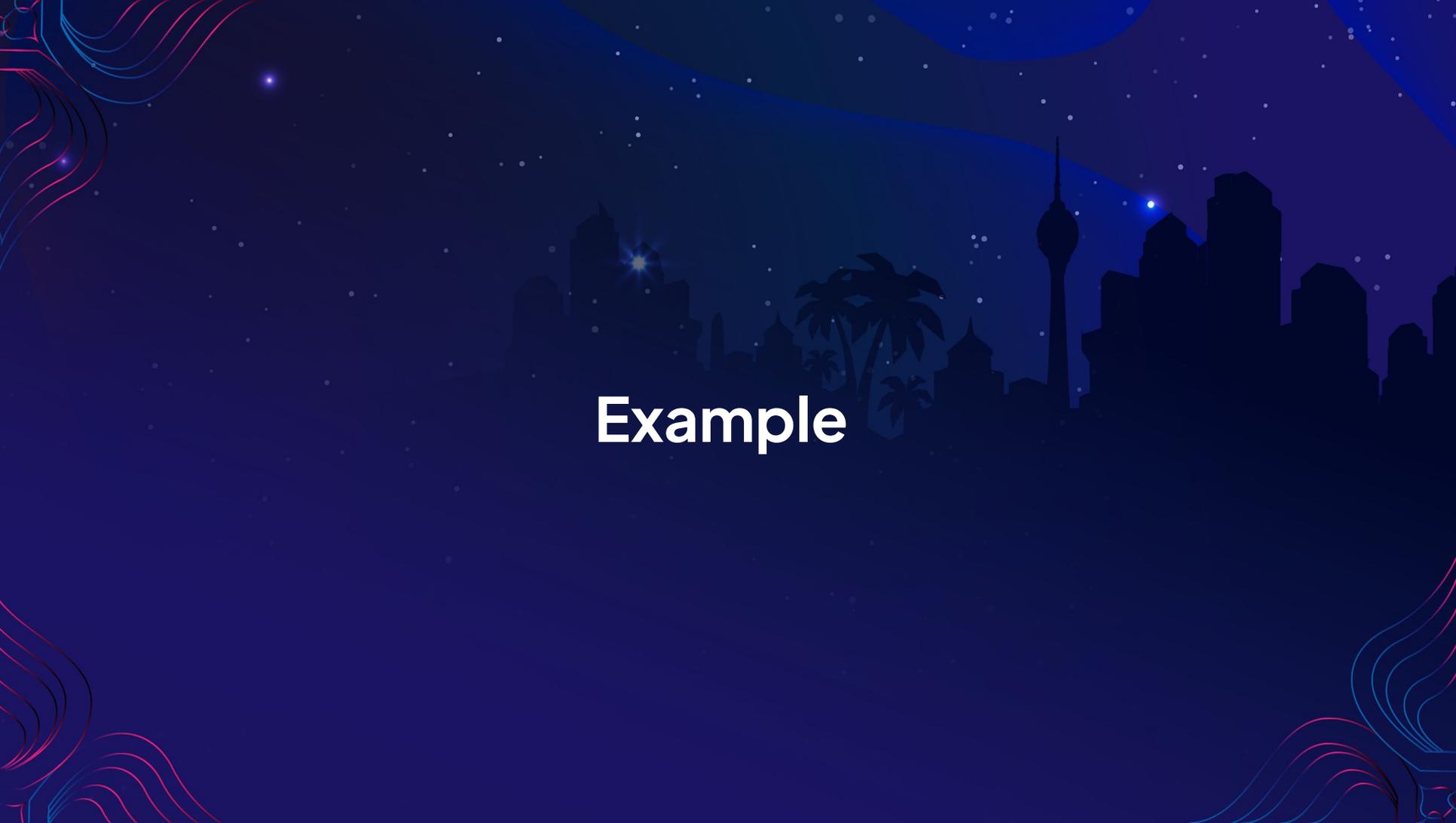


AI's biggest challenge isn't intelligence — it's integration.



Integration

Learn to unlearn and relearn

The image features a dark blue background with a city skyline silhouette. The skyline includes various buildings, palm trees, and a prominent tower with a spherical top. The sky is filled with small white stars and a few larger, brighter stars. In the corners, there are decorative wavy lines in shades of blue and pink. The word "Example" is centered in the middle of the image in a white, bold, sans-serif font.

Example

Online Retail Shop



Online Retail Shop



WSO2 Integrator: BI

The screenshot displays the WSO2 Integrator: BI interface. On the left is a navigation tree for the 'inventory_backend' project, showing various components like Entry Points, HTTP Services, Listeners, Connections, Types, Functions, Data Mappers, Configurations, and Local Connectors. The main workspace shows a diagram titled 'AI Chat Agent' with the following details:

- Start** node connects to the **AI Agent** component.
- AI Agent** component:
 - Input: `chat (@http:Payload ai:ChatReqMessage request)`
 - Output: `ai:ChatRespMessage|error`
 - Memory**: `ai:MessageWindowChatMemory`
 - Customer Support Representative** role description: "You are a helpful, friendly, and knowledgeable customer support agent for a clothing store. Your primary responsibility is to respond to customer queries promptly and accurately, ensuring a positive shopping experience. **Your responsibilities include:** 1. **Answering Questions About Products:** Provide information about sizes, colors, materials, availability, and style recommendations. * Help customers understand size charts and fit guides. 2. **Order and Delivery Inquiries:** Assist with order tracking, delivery timelines, shipping methods, and costs. * Update customers on order status or delays. 3. **Returns and Exchanges:** Explain return and exchange policies. * Guide customers through the return/exchange process. 4. **Issue Resolution:** Handle complaints or issues related to damaged, incorrect, or missing items. * Escalate issues to a supervisor if needed. 5. **Store and
 - Functions (indicated by dashed lines):
 - `listAllClothingProdu...`
 - `createNewProduct` (highlighted with a blue circle)
 - `getProductDetails`
 - `updateProductDetails`
 - `removeProduct`
 - `updateQuantity`
- Return** node: `{message: stringResult}`

The bottom status bar shows 'Ballerina 2201.12.7 (Swan Lake Update 12)'.

WSO2 Integrator: BI

The screenshot displays the WSO2 Integrator: BI interface, divided into three main sections:

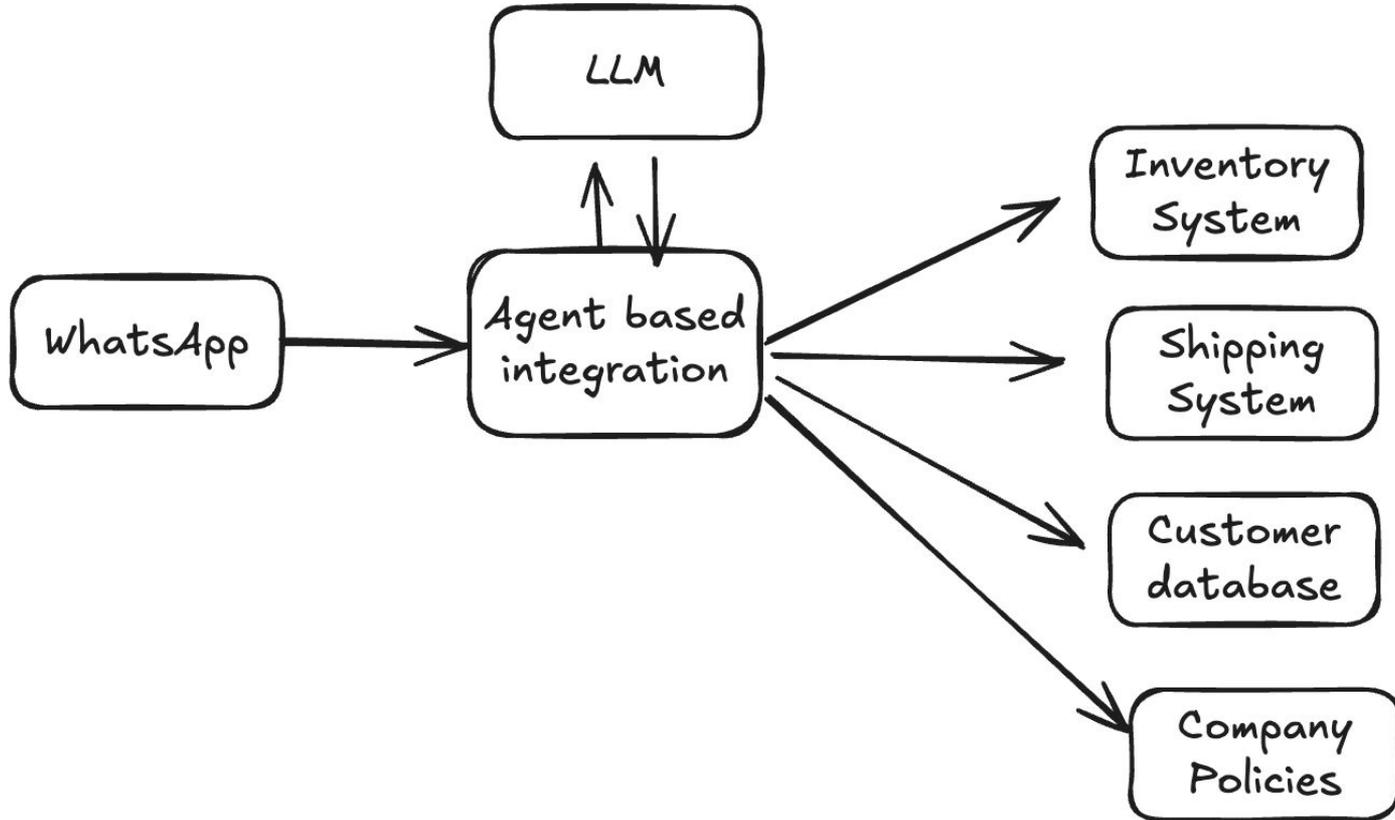
- Left Panel (Project Explorer):** Shows the project structure for 'inventory_backend'. It includes 'Entry Points' (AI Agent Services, HTTP Service), 'Listeners' (inventory_agentListener), 'Connections' (_inventory_agentModel, inventoryClient), 'Types' (Category, Cur, Category ry, ErrorResponse, InventoryBySize, InventoryUpdate, Product, ProductCreate, ProductUpdate, Size), 'Functions' (createNewProduct, getProductDetails, initializeMockData, listAllClothingProducts, removeProduct, updateProductDetails, updateQuantity), 'Data Mappers', 'Configurations', and 'Local Connectors'.
- Center Panel (Service Designer):** Displays a diagram for the 'AI Chat Agent' service. The diagram starts with a 'Start' node leading to an 'AI Agent' function block. This block contains a 'Memory' section with 'ai:MessageWindowChatMemory'. Below the memory is a 'Customer Support Representative' role description. The diagram then branches into several function calls: 'listAllClothingProdu...', 'createNewProduct', 'getProductDetails', 'updateProductDetails', 'removeProduct', and 'updateQuantity'. The flow concludes with a 'Return' node containing 'message: stringResult'.
- Right Panel (Agent Chat):** Shows a chat interface with the following messages:
 - Agent: "Hello! How can I assist you today?"
 - User: "I would like to know the available products"
 - Agent: "Could you please specify the category of clothing you're interested in, along with any preferred material or size? This will help me provide you with the most relevant options."
 - User: "can you list down the categories"
 - Agent: "Sure! Here are some common clothing categories:
1. **Tops** (e.g., t-shirts, blouses, sweaters)
2. **Bottoms** (e.g., jeans, skirts, shorts)
3. **Dresses** (e.g., casual, formal, maxi)
4. **Outerwear** (e.g., jackets, coats, hoodies)
5. **Activewear** (e.g., leggings, sports bras, workout tops)
6. **Loungewear** (e.g., pajamas, sweatpants, comfy tops)
7. **Accessories** (e.g., hats, scarves, belts)
8. **Footwear** (e.g., sneakers, boots, sandals)
Let me know which category you're interested in, and if you have a specific material or size preference!"
 - User: "Tops"
 - Agent: "Great choice! Do you have any specific preferences for material (like cotton, polyester, etc.) or size? This will help me narrow down the available options for tops."
 - User: "cotton"
 - Agent: "Could you please specify your preferred size for the cotton tops? This will help me provide you with the most accurate options!"

Great, so where is the challenge...

AI's biggest challenge isn't intelligence — it's integration.



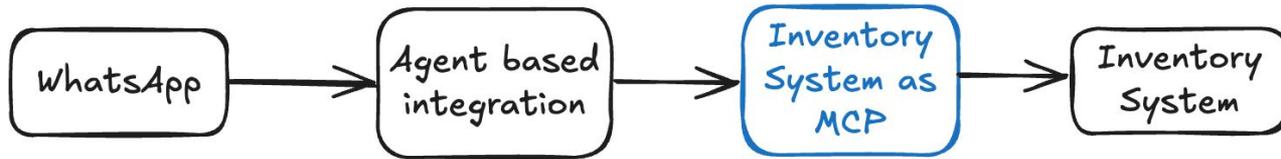
Online Retail Shop





MCP

Online Retail Shop



BI MCP Support

The screenshot displays the WSO2 Integrator BI interface. The main workspace shows a diagram for an **AI Chat Agent** with the following components:

- Start** node leading to the **AI Agent** component.
- AI Agent** component with properties: `stringResult`, `Memory` (value: `ai:MessageWindowChatMemory`), and a description: "Weather AI Assistant. You are Nova, a smart AI assistant dedicated to providing accurate and timely weather information. Your primary responsibilities include: - Current Weather: Provide detailed and user-friendly current weather information for a given location."
- mcpServer** component connected to the AI Agent.
- Return** node with properties: `{message: stringResult}`.

On the right, the **Add MCP Server** configuration panel is open, showing the following settings:

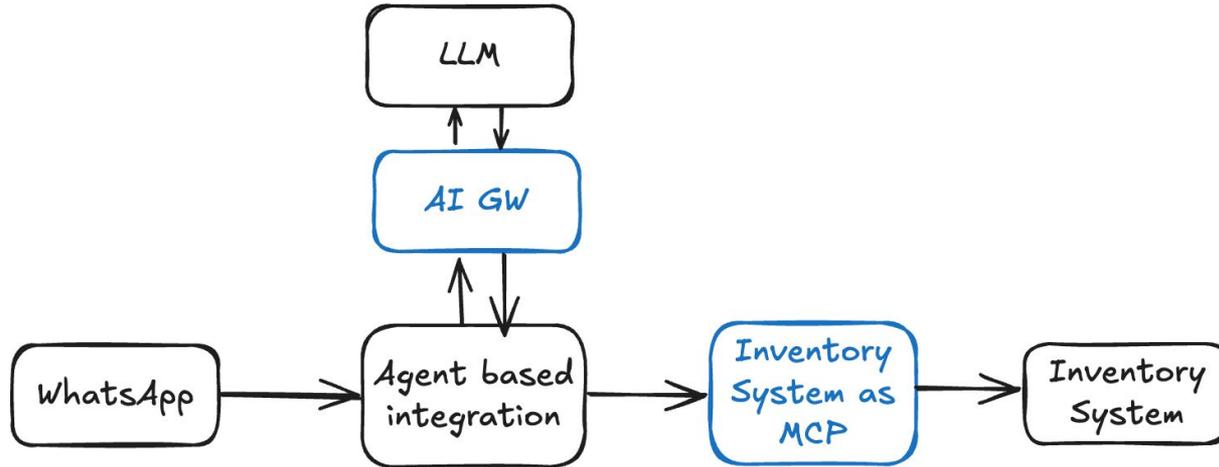
- Server Uri***: MCP server URL. Example: `"http://localhost:3000/stream"`
- Info***: Client information such as name and version. Example:

```
{  name: "Weather MCP Server",  version: "1.0"}
```
- Tools To Include**: Select the tools to include in the MCP server. A dropdown menu shows "Selected".
- Available Tools**: A list of tools with checkboxes, including `get-current-weather`, `get-weather-forecast`, `get-hourly-forecast`, `get-daily-forecast`, and `get-minutely-forecast`. A **Deselect All** button is present.
- Optional Configurations**: A section with an **Expand** dropdown.

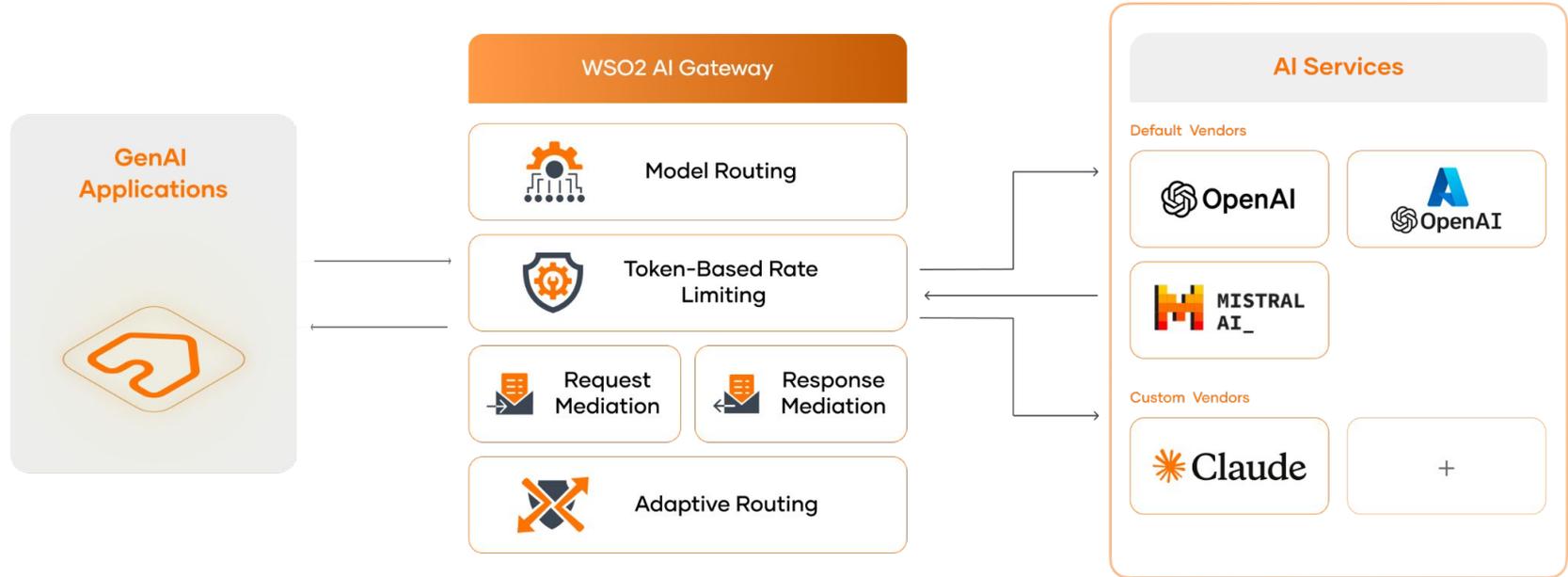


Governance

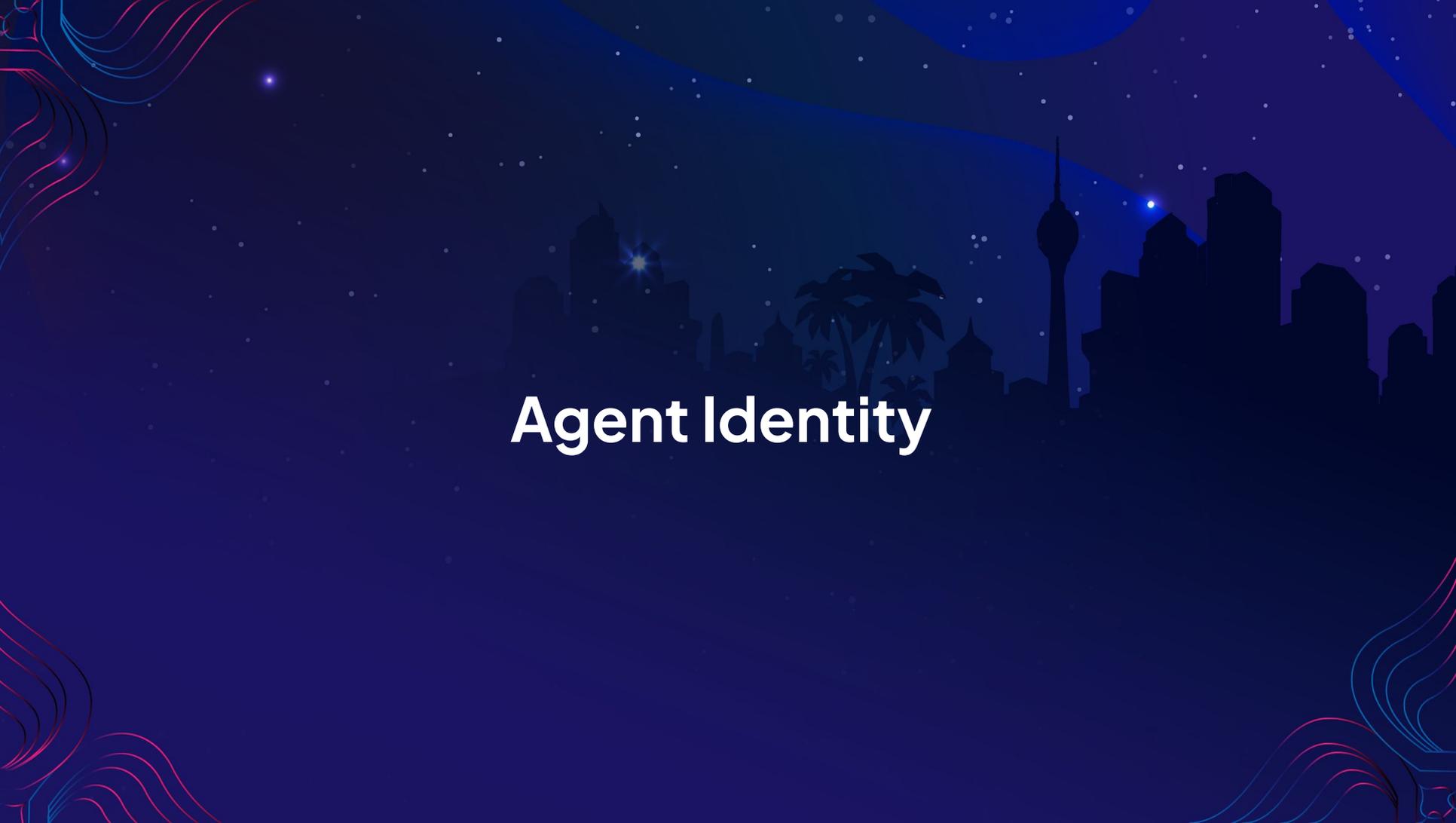
Online retail shop



WSO2 AI Gateway

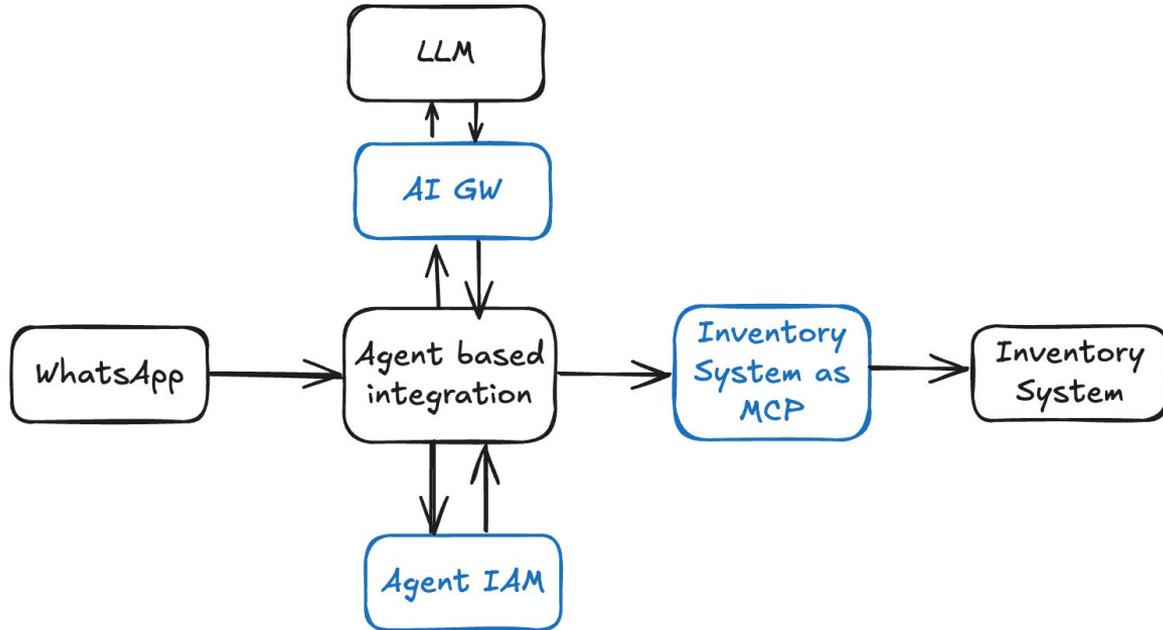


Guardrail	Sample Use Case
Semantic Prompt Guard	Stops prompts like “Write my homework” in a student assistant app.
Regex based PII Masking	Masks credit card numbers in user prompts for a payment chatbot.
Word Count and Sentence Count	Limits AI replies to 50 words in a quick-answer mobile app
JSON Schema Validation	Validates API responses for correct format in an e-commerce platform
Regex Validation	Verifies user-entered email addresses in a registration form
URL Validation	Ensures links in AI responses resolve via DNS for a news aggregator app.
Content Length	Caps user inputs at 500 characters in a chat AI to prevent spam
Grounded AI Hallucination	Prevents AI from making up facts in product descriptions.
Content Safety	Filters hate speech in comments generated by an AI writing assistant.
PII Detection and Masking	Detects and hides phone numbers in support chatbot inputs.
Jailbreak detection	Stops prompts like “Ignore all rules” in customer service bots.



Agent Identity

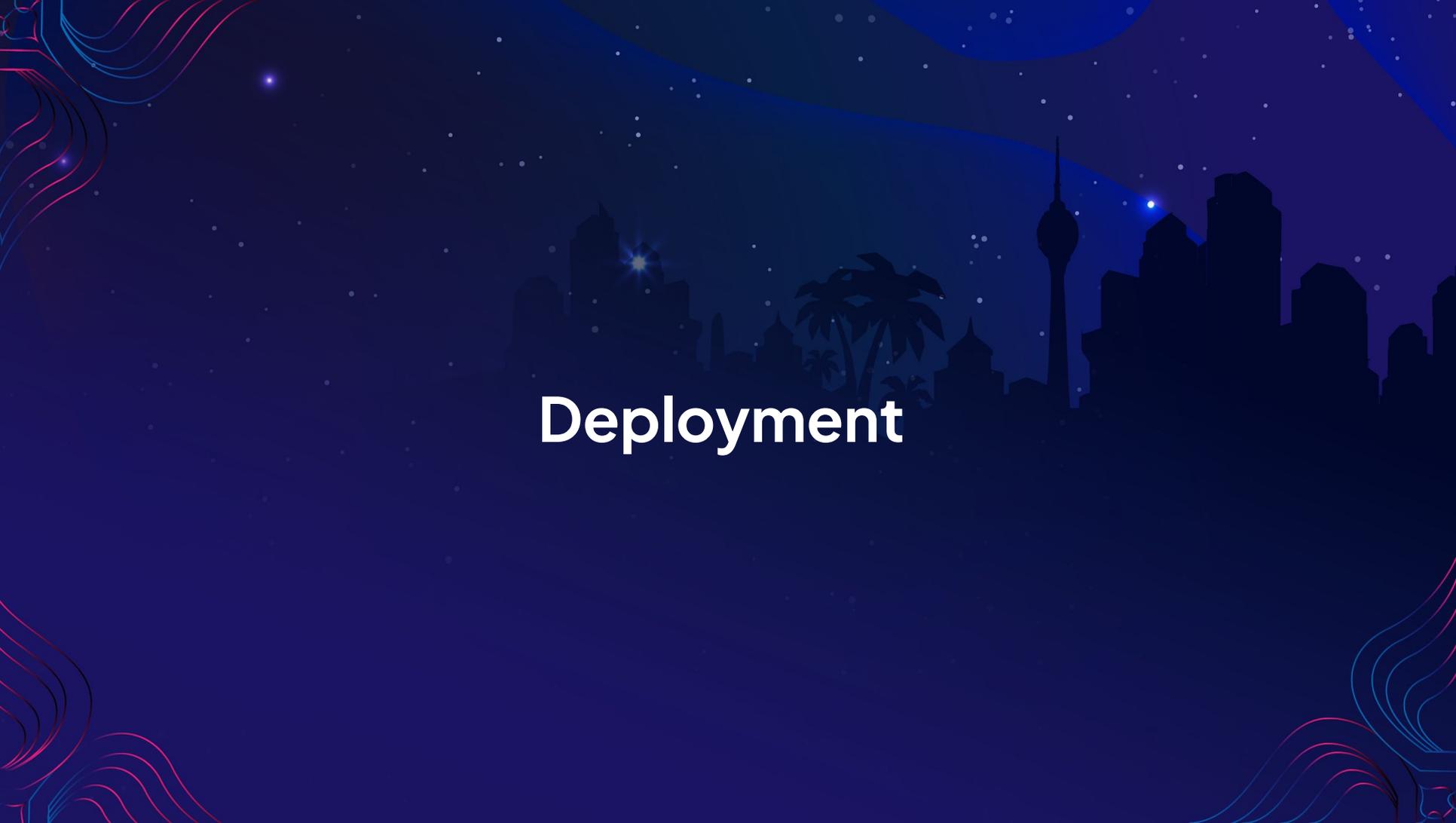
Online retail shop



Asgardeo Agent Identity

- **Verify Identity**
 - ⦿ Every AI agent must authenticate and maintain identity
 - ⦿ Continuous validation of agent legitimacy
- **Just in time / Just enough access:**
 - ⦿ Agents receive minimum permissions for their specific tasks
 - ⦿ Dynamic permission adjustment based on context
- **Assume Breach**
 - ⦿ Design systems expecting agent compromise
 - ⦿ Limit blast radius of potential security incidents
- **Continuous Monitoring**
 - ⦿ Real-time audit of agent actions and decisions
 - ⦿ Behavioral analysis for anomaly detection





Deployment

Devant

The screenshot displays the Devant IDE interface for a Ballerina project named "WSO2 Integrator: BI". The left sidebar shows a project tree with folders for "Entry Points", "Listeners", "Connections", "Types", "Functions", "Data Mappers", "Configurations", and "Local Connectors". The main workspace shows a diagram for an "AI Chat Agent" with the following components:

- Start**: The entry point of the agent.
- AI Agent**: A component that returns a `stringResult`. It includes a **Memory** section with the type `ai:MessageWindowChatMemory` and a **Define agent's role** section with the instruction: "Provide specific instructions on how the agent should behave."
- Return**: A component that returns a `message: stringResult`.

The diagram is connected to a **Configure Memory** panel on the right, which allows selecting a memory manager from a list of available options.

At the bottom of the IDE, the status bar shows "WSO2 Ballerina 2201.12.7 (Swan Lake Update 12) Layout: U.S."

Devant

The screenshot displays the Devant web interface for configuring an AI Agent. At the top, the navigation bar includes the Devant logo, organization name 'shafreenanfar', project 'hr-chatbot', and integration 'bi-hr-agent'. A 'Developer (Default)' dropdown and an 'Upgrade' button are also present.

The left sidebar contains navigation options: Overview, Develop, Build, Deploy, Test, Insights, Observability, and Admin.

The main content area is titled 'Deployment Track' and shows the 'bi-hr-agent' AI Agent. It includes options to 'Develop in VS Code' and 'Developer Portal'. The agent's status is 'Published' under 'Lifecycle Status'. A 'Compliance Summary' section indicates 'No Policies Applied'.

The 'Development' section shows the agent is 'Active' and provides a 'Configure' button. The 'Base URL' is displayed as `https://d1e273de-8e44-4809-91a6-5f67c76aff34-dev.e1-us-east-azure.choreoapis.dev/hrchatbot/bi-hr-agent/v1.0`. A chat interface shows a user message 'Hi' and an assistant response 'Hello! How can I assist you today?'. A 'Send' button is located at the bottom right of the chat area.

Devant

The screenshot displays the Devant web interface. At the top left is the Devant logo and the organization name 'shafreenanfar'. A navigation sidebar on the left lists: Overview, Develop, Build, Deploy, Test, Insights, Observability, and Admin. The main content area is titled 'Setup RAG Ingestion' and shows a progress bar with five steps: 1. Initialize Vector Store (active), 2. Configure Embedding Model, 3. Configure Chunking, 4. Upload Files, and 5. Verify. Below the progress bar, the 'Initialize Vector Store' step is expanded to show five options: Pinecone, Chroma, Weaviate, PostgreSQL, and Devant Managed. At the bottom of this section are 'Back' and 'Next' buttons.

Organization: shafreenanfar

Develop

- Overview
- Develop
- Build
- Deploy
- Test
- Insights
- Observability
- Admin

Setup RAG Ingestion

Initialize Vector Store

- Pinecone
- Chroma
- Weaviate
- PostgreSQL
- Devant Managed

Back Next

STEP 1 Initialize Vector Store

STEP 2 Configure Embedding Model

STEP 3 Configure Chunking

STEP 4 Upload Files

STEP 5 Verify

A dark blue background featuring a silhouette of a city skyline with various skyscrapers and palm trees. The sky is filled with small white stars and a few larger, brighter stars. In the corners, there are decorative wavy lines in shades of blue and pink. The text "Evaluations (WIP)" is centered in white.

Evaluations (WIP)

Right set of tools with wrong set of integrations...



Autonomy

Coldplay moment

- Agentic Misalignment



Popular Agents

Customer
Support Agent

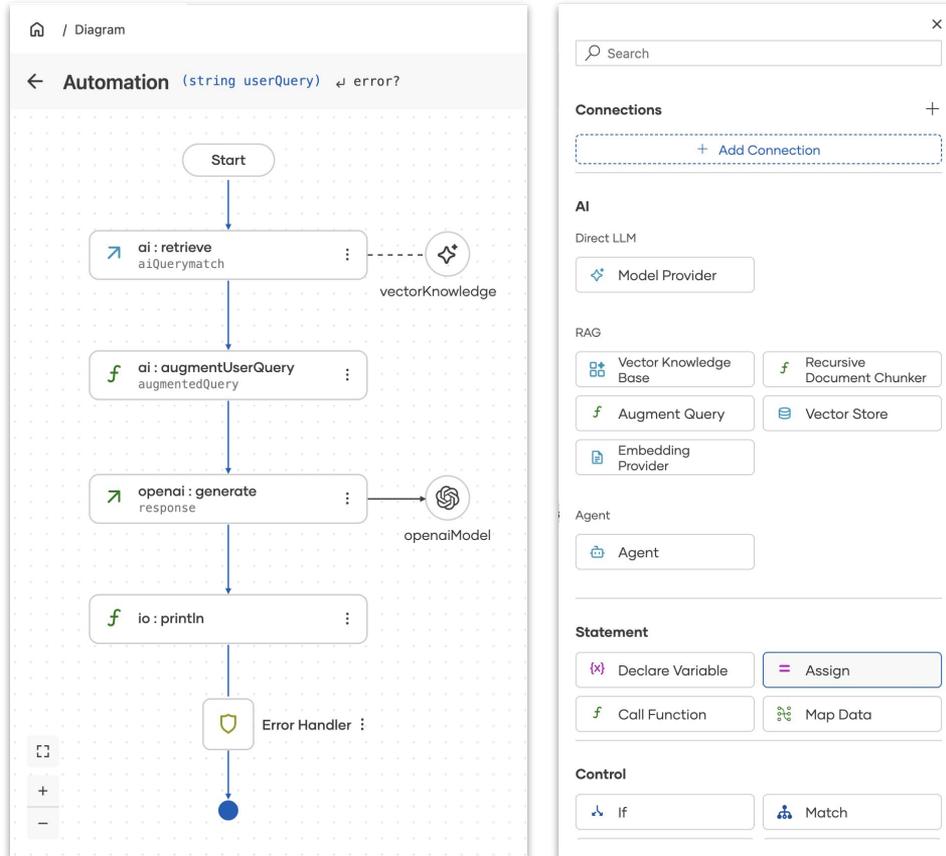
Employee
Support Agent

Doctor
Appointment
Agent

Parking Query
Agent

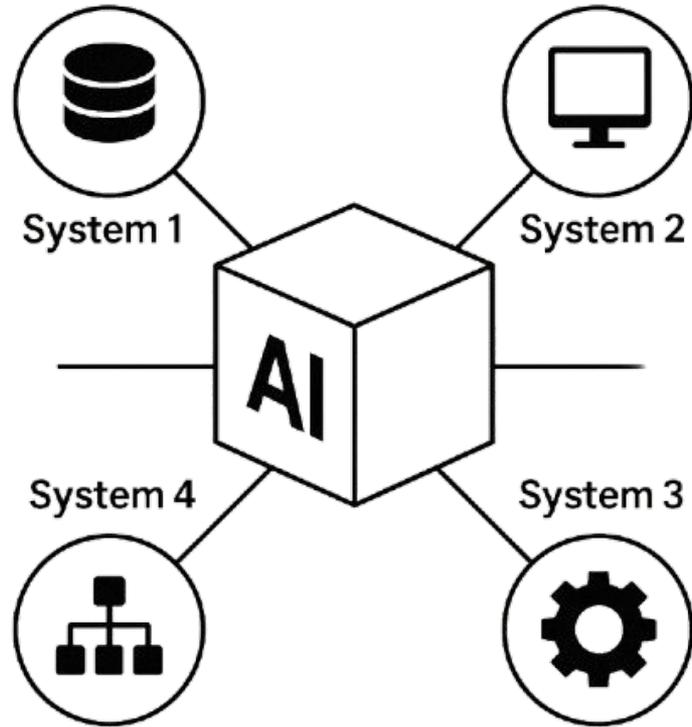
Retail
Assistant
Agent

Simpler Agents



Key Takeaways

AI's biggest challenge isn't intelligence — it's integration.



改善



Question Time!





Thank you!

20TH ANNIVERSARY EDITION

WSO2CONASIA

—  PLATFORMLESS MODERNIZATION