
# TUNISIA AI TOUR 2026

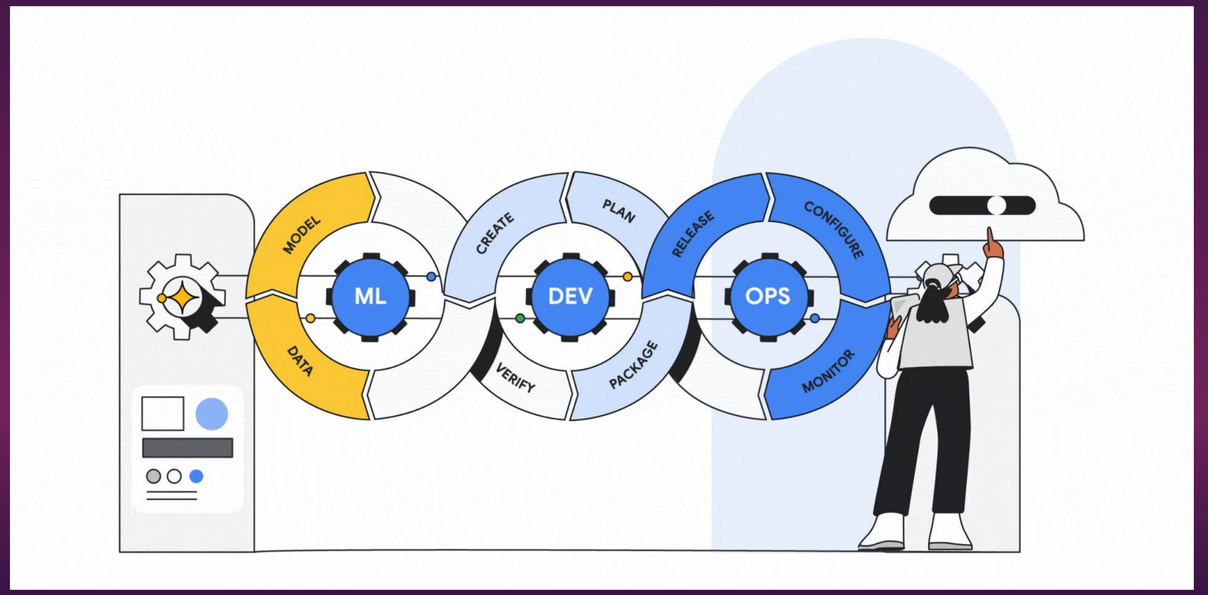

- Senior AI Engineer , Netflix
- AI Expert Consultant , Tegus
- PHD Candidate in xAI ,Saint Mary's University Canada
- CEO and Head of AI R&D ,Man.AI
- International AI Speaker (Linkedin AI Top Voice)
- AI Expert Reviewer & Author , Packt Publishing UK


# Building Agentic AI Workflows with xAI on Google Vertex AI: " From Code to Action "

**Exploring the Development and Implementation of Autonomous AI Workflows with xAI layers  on Google Cloud Vertex AI.**

# Session Goals

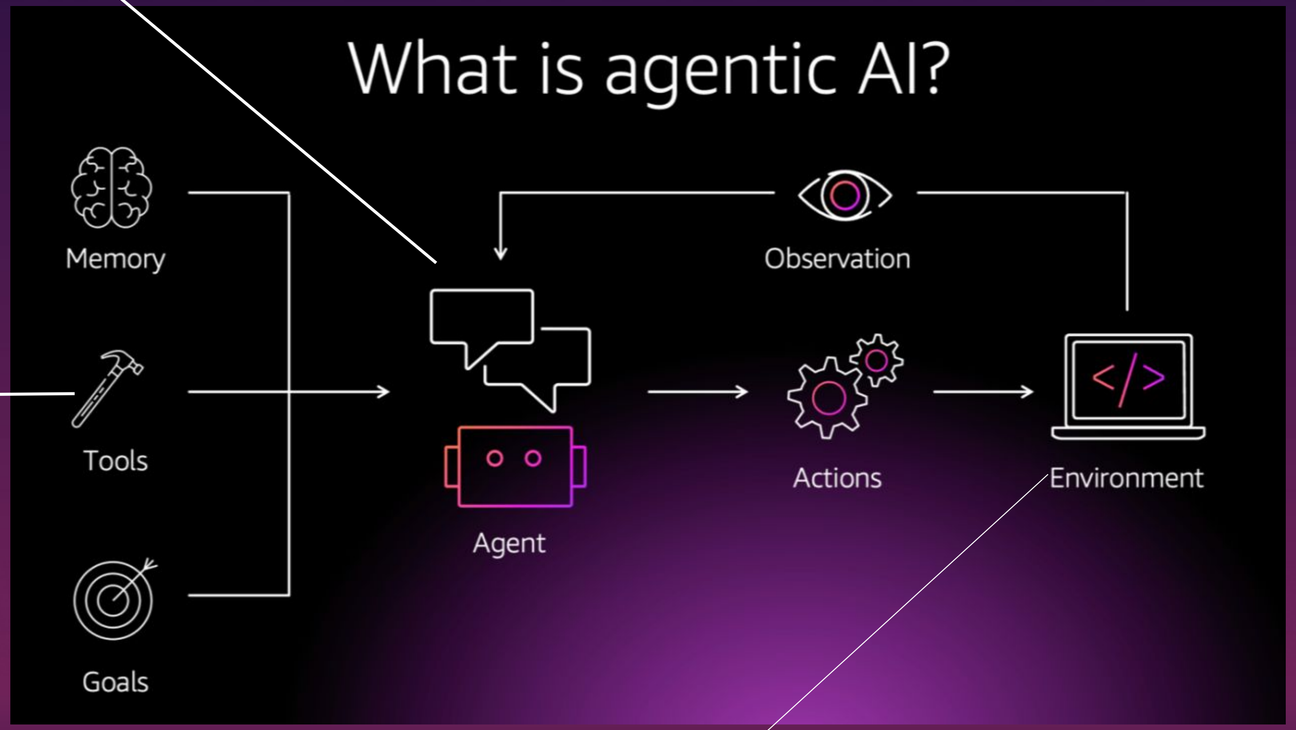
- Understand agentic AI & xAI principles
- Learn to implement end-to-end workflows on Vertex AI
- See live Python code for autonomous agents and how to Deploy it on Google Vertex AI



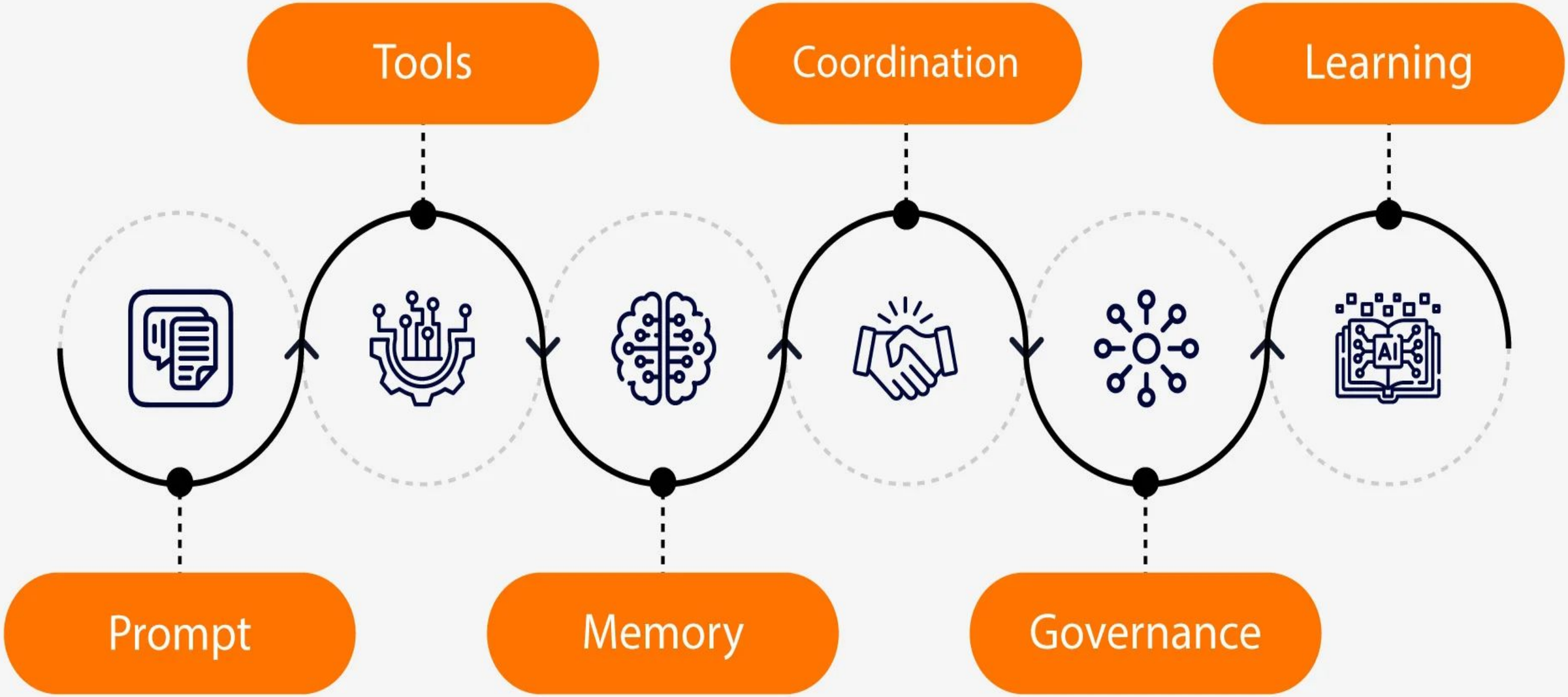
# What is Agentic AI

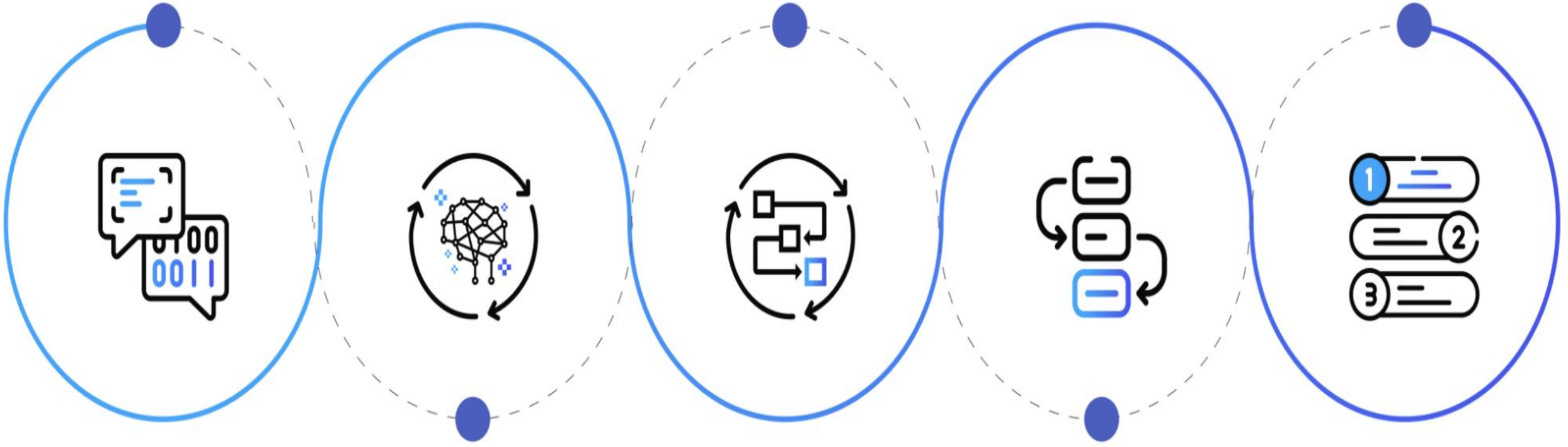

**Autonomous systems** that can operate independently without constant human supervision.

These systems are designed to perform specific tasks by making decisions based on their learned experiences and data inputs.




# How Agentic AI Really works ?


Tools
Coordination
Learning
AI
Prompt
Memory
Governance


Natural Language Input
Interpretation & Reasoning
Workflow Generation
Workflow Execution
Output/ Outcome
Learning & course correction
(eg. through reinforcement learning)

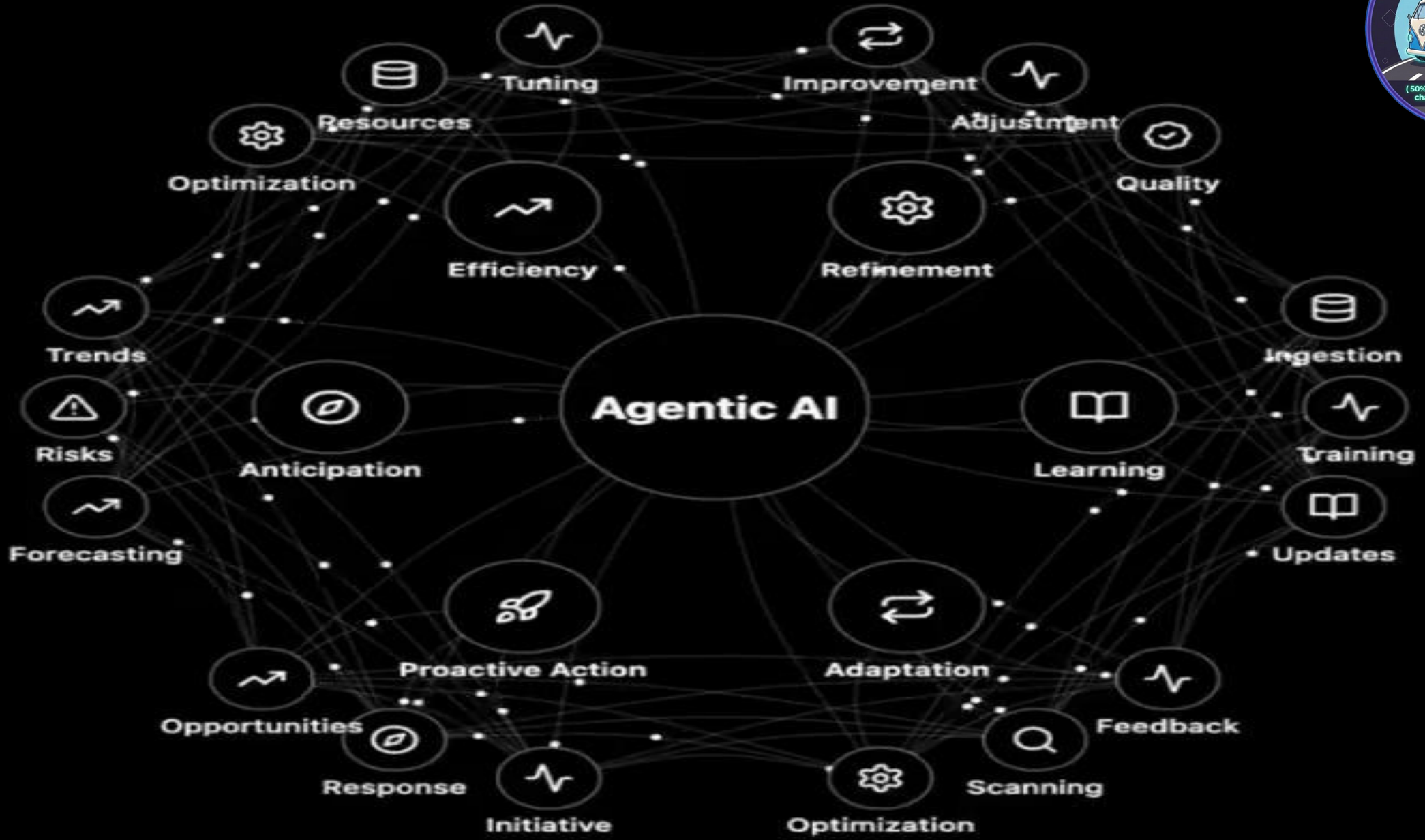
# KEY TAKEAWAY N°1

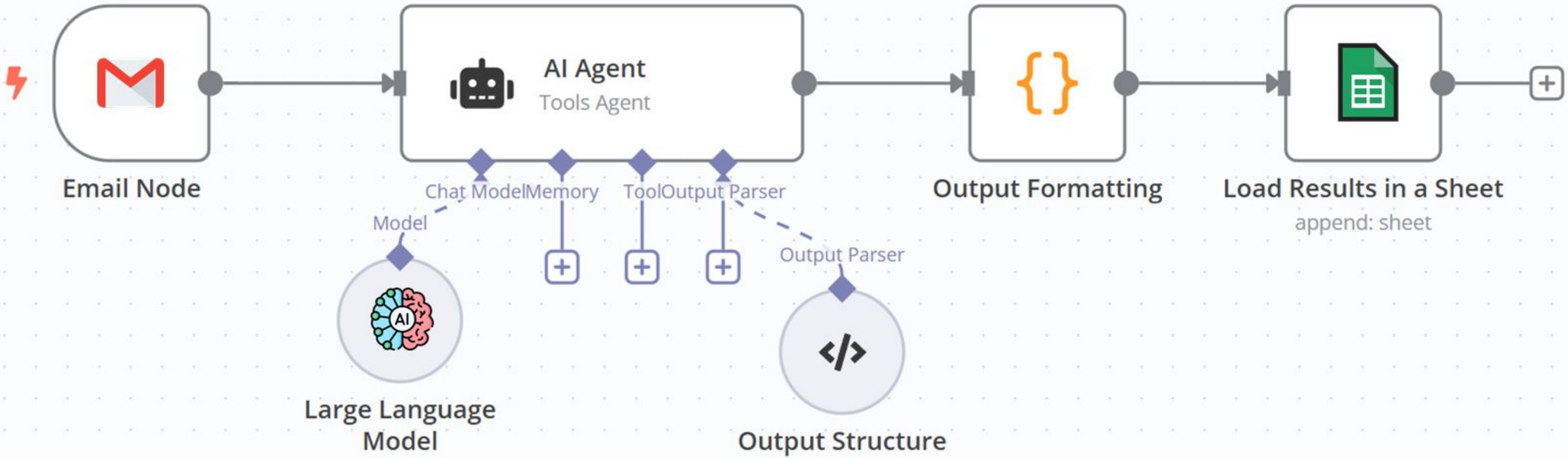

**Agentic AI works by taking user goals and autonomously using a cycle of perception, reasoning, action, and reflection to achieve them, unlike traditional AI that simply reacts to commands.**

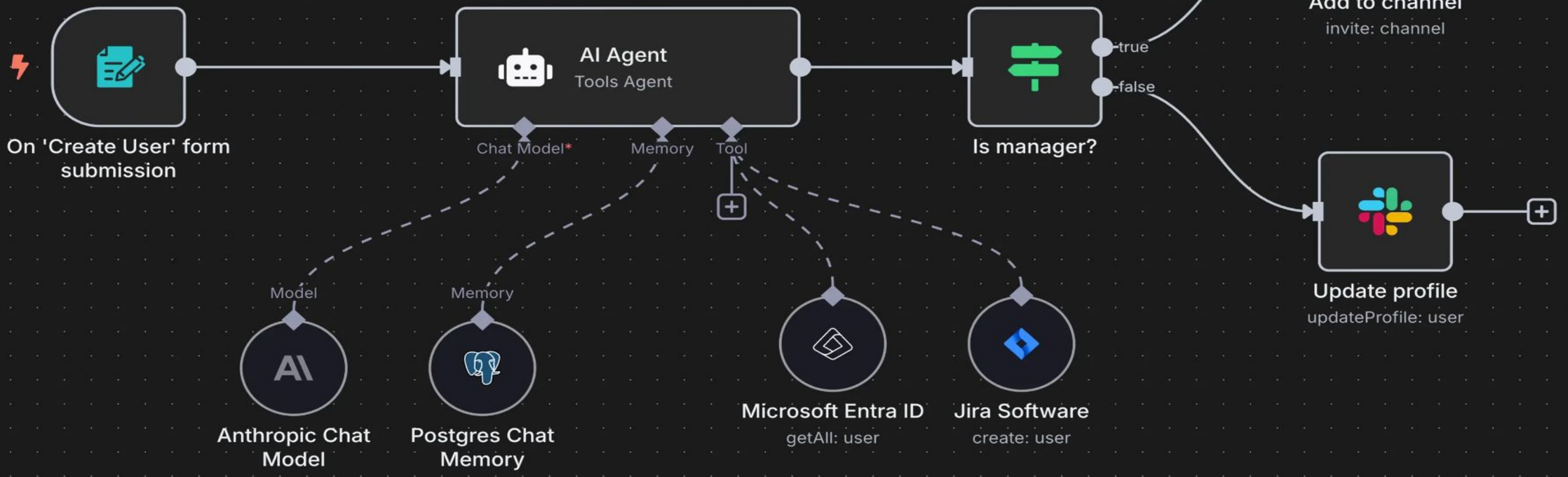
**It gathers data, uses a Large Language Model (LLM) to understand the context and plan a strategy, executes tasks by interacting with other systems or APIs, and then reflects on the outcomes to improve its future actions.**

**This makes it capable of handling complex, multi-step tasks independently and proactively.**

# How Agentic AI Works (Proactive & Strategic)




AI Agent
Tools Agent
Email Node
Chat Model
Memory
Tool
Output Parser
Model
Output Parser
Large Language Model
Output Structure
Output Formatting
Load Results in a Sheet
append: sheet


On 'Create User' form submission
AI Agent
Tools Agent
Chat Model*
Memory
Tool
Is manager?
true
false
Add to channel
invite: channel
Update profile
updateProfile: user
Model
Anthropic Chat Model
Memory
Postgres Chat Memory
Microsoft Entra ID
getAll: user
Jira Software
create: user


# What is Explainable AI (xAI)?

## Today

Training Data → Machine Learning Process → Learned Function

Learned Function → Decision or Recommendation → User

Task → User

- Why did you do that?
- Why not something else?
- When do you succeed?
- When do you fail?
- When can I trust you?
- How do I correct an error?

## XAI

Training Data → New Machine Learning Process → Explainable Model → Explanation Interface

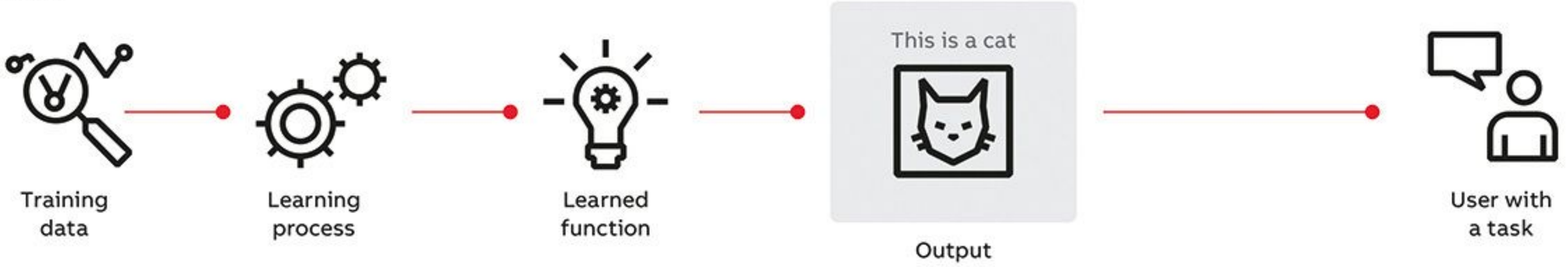
Explanation Interface ⇄ User

Task → User

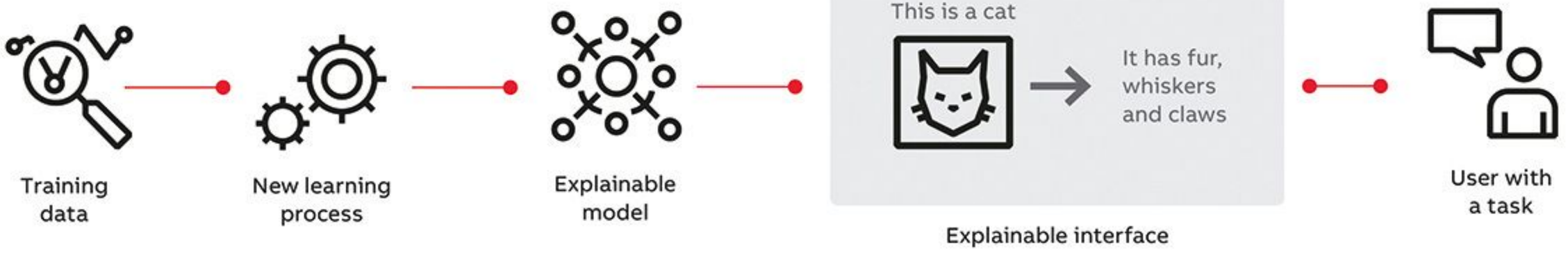
- I understand why
- I understand why not
- I know when you succeed
- I know when you fail
- I know when to trust you
- I Know why you erred

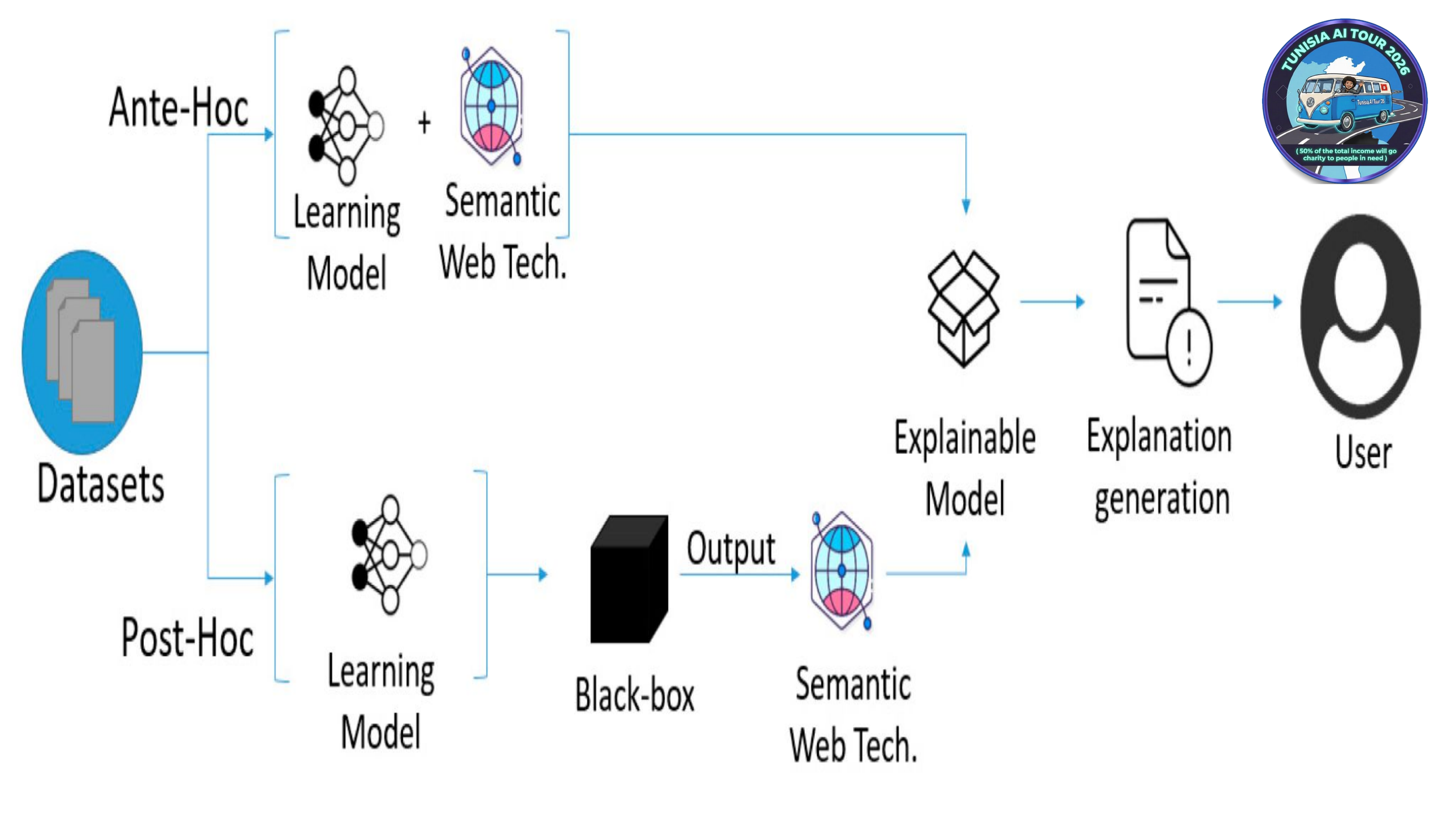

# How Explainable AI Really works ?

**TODAY**




**TOMORROW**




Ante-Hoc
Learning Model
+
Semantic Web Tech.
Datasets
Post-Hoc
Learning Model
Black-box
Output
Semantic Web Tech.
Explainable Model
Explanation generation
User

# KEY TAKEAWAY N°2


Explainable AI (XAI) refers to methods and techniques that help humans understand why an AI system made a specific decision or prediction, moving beyond the "black box" of complex models like deep learning.

It provides transparency into the AI's reasoning, highlighting key factors, potential biases, and algorithms used, which builds trust, helps debug models, ensures fairness, and meets regulatory needs in critical fields like healthcare or finance.

# Key Features of Explainable AI


## Fairness and Bias Detection

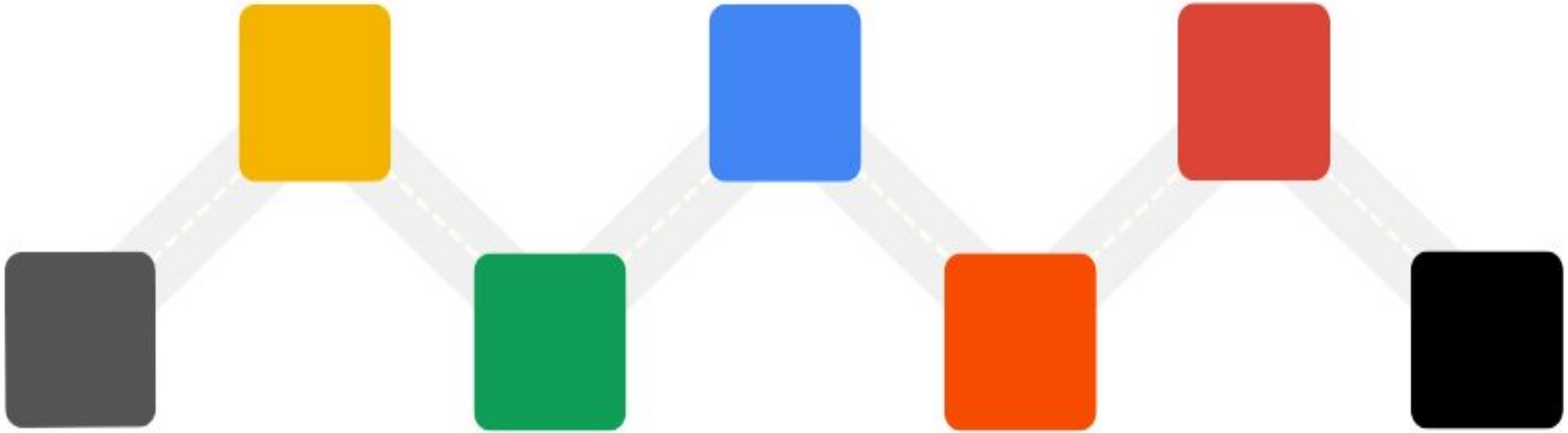
Identifies and mitigates biases in AI models

## User Control and Trust

Enhances user confidence by offering explanations

## Adversarial Robustness

Resists misleading inputs and adversarial attacks



## Transparency

Clear insights into how the model makes decisions

## Interpretability

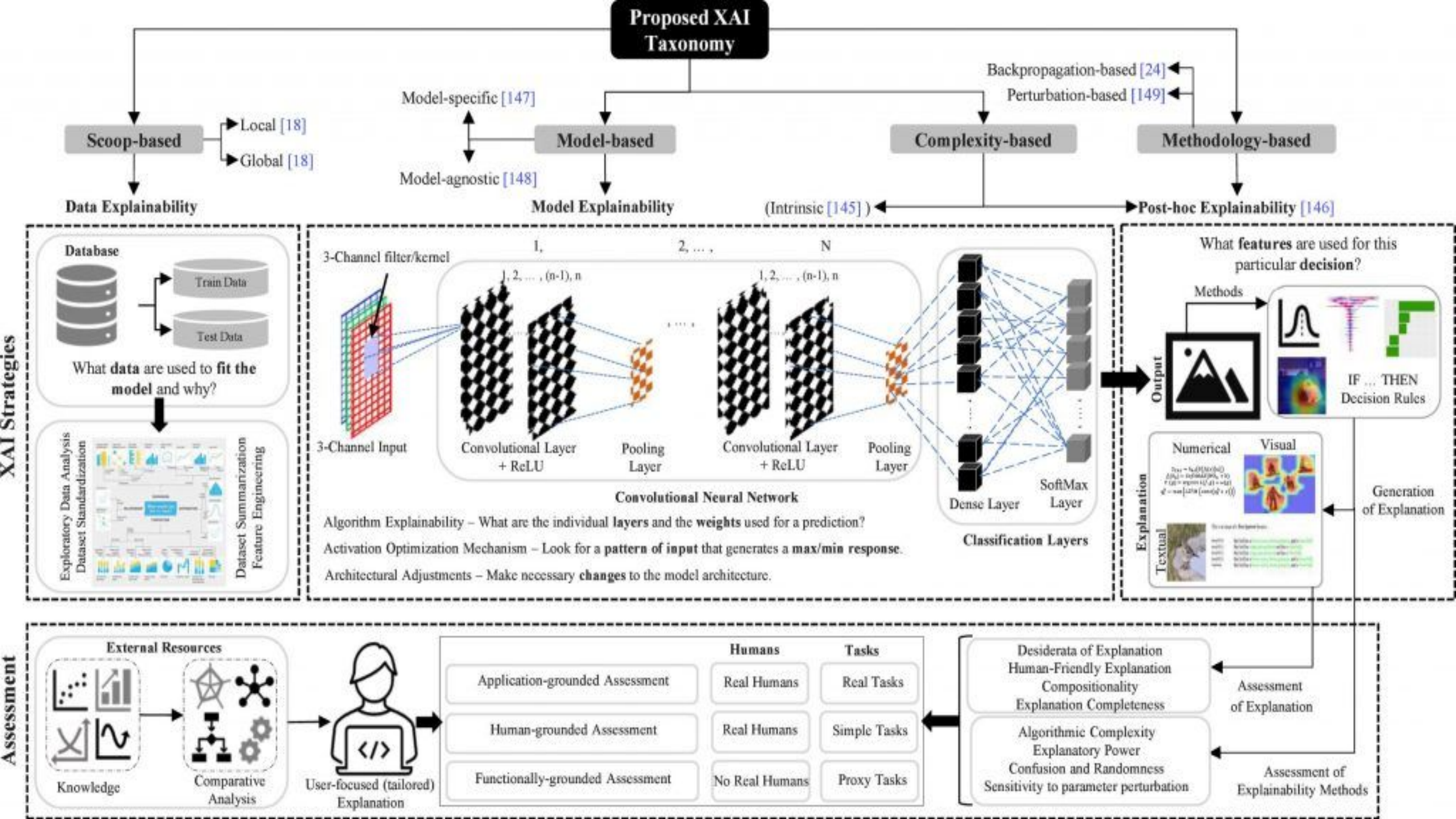
AI outputs can be easily understood by humans

## Traceability

Ability to track and audit the decision making process

## Justification

AI provides reasons or evidence for its predictions


Proposed XAI Taxonomy
Scoop-based
Local [18]
Global [18]
Data Explainability
Model-specific [147]
Model-based
Model-agnostic [148]
Model Explainability
Backpropagation-based [24]
Perturbation-based [149]
Complexity-based
(Intrinsic [145] )
Methodology-based
Post-hoc Explainability [146]
XAI Strategies
Database
Train Data
Test Data
What data are used to fit the model and why?
Exploratory Data Analysis
Dataset Standardization
Dataset Summarization
Feature Engineering
3-Channel filter/kernel
1,
2, … ,
N
1, 2, … , (n-1), n
3-Channel Input
Convolutional Layer + ReLU
Pooling Layer
Convolutional Neural Network
Dense Layer
SoftMax Layer
Classification Layers
Algorithm Explainability – What are the individual layers and the weights used for a prediction?
Activation Optimization Mechanism – Look for a pattern of input that generates a max/min response.
Architectural Adjustments – Make necessary changes to the model architecture.
What features are used for this particular decision?
Methods
Output
IF … THEN Decision Rules
Explanation
Numerical
Visual
Textual
Generation of Explanation
Assessment
External Resources
Knowledge
Comparative Analysis
User-focused (tailored) Explanation
Humans
Tasks
Application-grounded Assessment
Real Humans
Real Tasks
Human-grounded Assessment
Real Humans
Simple Tasks
Functionally-grounded Assessment
No Real Humans
Proxy Tasks
Desiderata of Explanation
Human-Friendly Explanation
Compositionality
Explanation Completeness
Assessment of Explanation
Algorithmic Complexity
Explanatory Power
Confusion and Randomness
Sensitivity to parameter perturbation
Assessment of Explainability Methods


TUNISIA AI TOUR 2026
Tunisia AI Tour '26
( 50% of the total income will go charity to people in need )

# Vertex AI

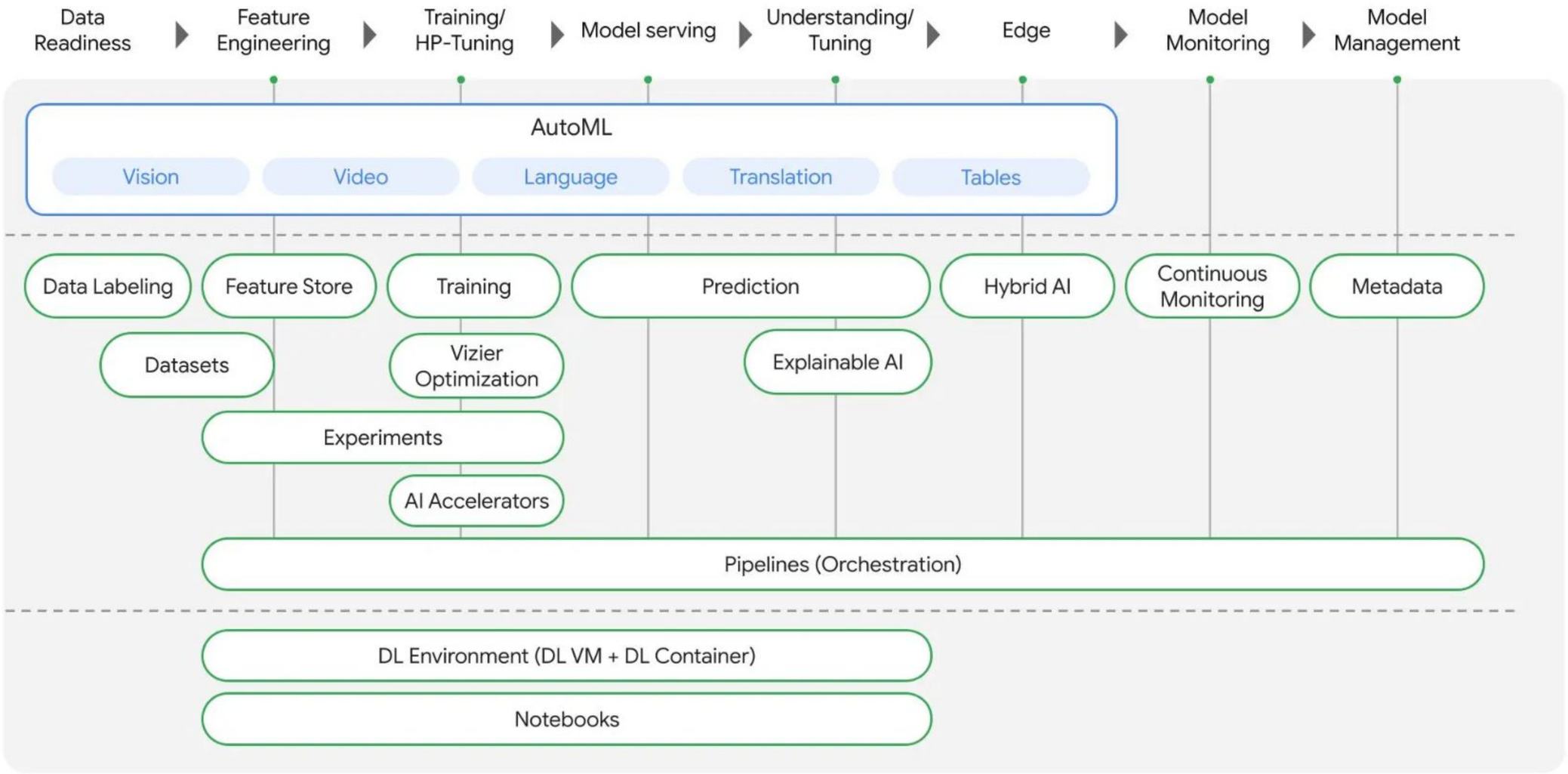
# Google Cloud Vertex AI Overview


**Google Cloud Vertex AI**

**is a fully-managed, unified platform for building, deploying, and scaling generative AI and machine learning (ML) models at an enterprise level.**

**It unifies Google Cloud's previously separate AI services (AutoML and AI Platform) into a single environment, designed to streamline the entire ML lifecycle.**

# What's included in Vertex AI?



# Innovate faster with enterprise-ready AI, enhanced by Gemini models

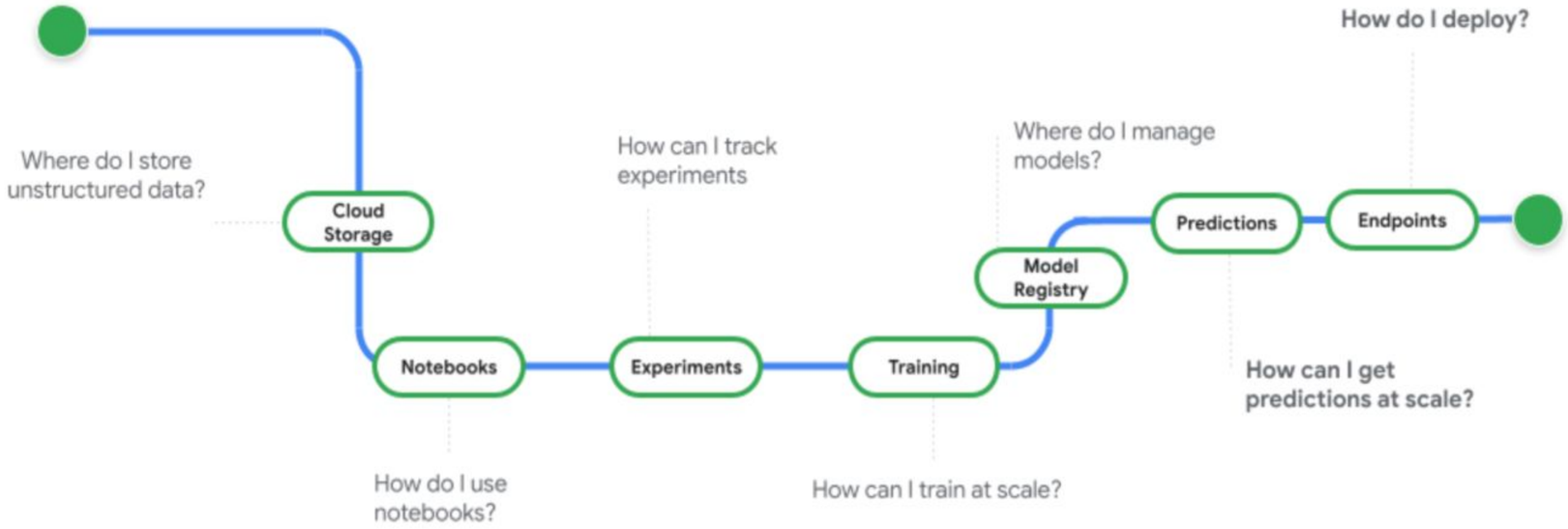
Vertex AI is a fully-managed, unified AI development platform for building and using generative AI. Access and utilize Vertex AI Studio, Agent Builder, and 200+ foundation models.

Try it in console | Contact sales

**Want training?** Start a free course for Vertex AI Studio.

Vertex AI Platform
Features
How It Works
Common Uses
Build with Gemini models
AI Agents and applications
Extract, summarize, and classify data
Train custom ML models
Deploy a model for production use
Generate a solution
Pricing
Business Case
Try Gemini in Vertex AI
Documentation
Sample code and

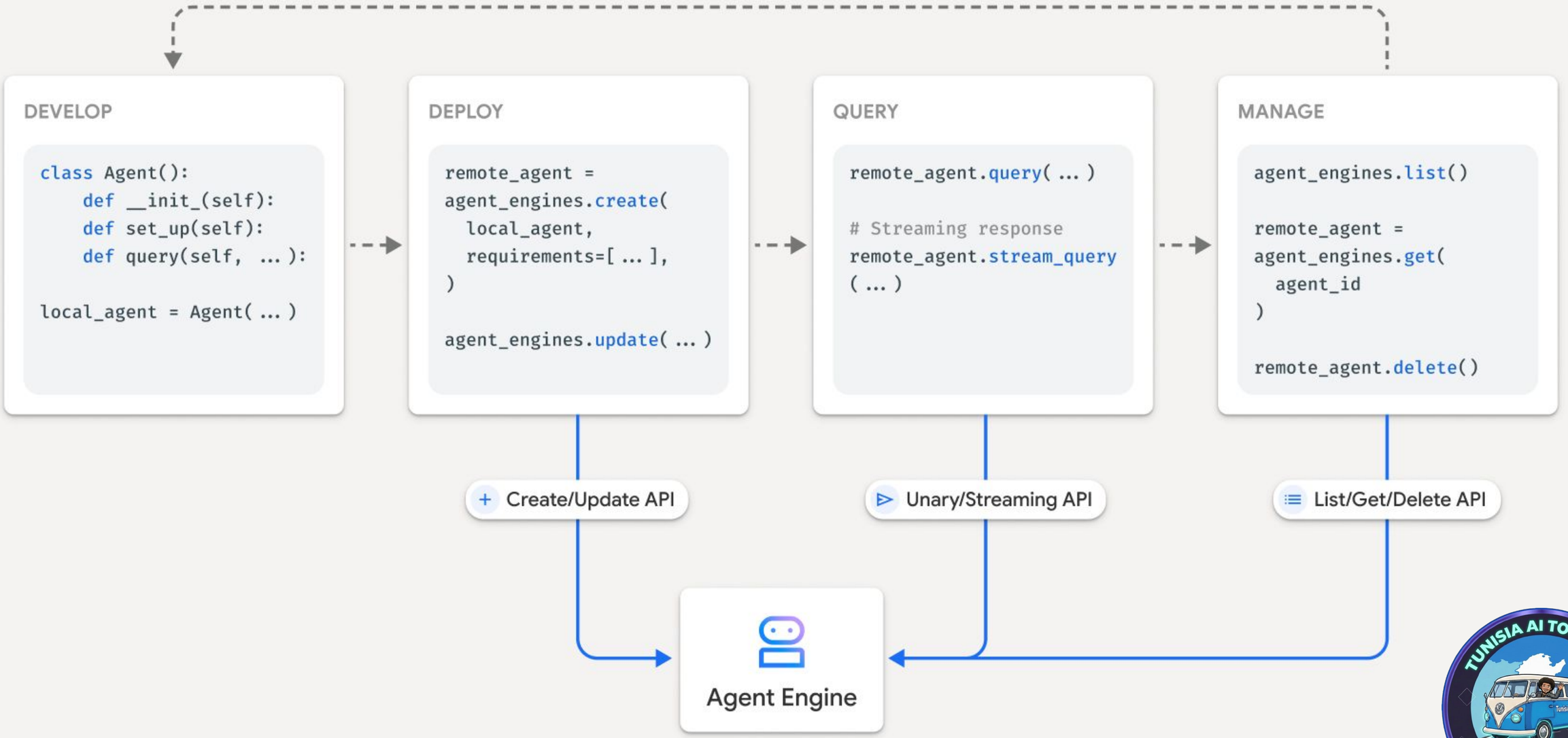
## The Prototype to Production Journey




Watch Prototype to Production, a video series that takes you from notebook code to a deployed model.


# Announcing Vertex AI Agent Builder

Google Cloud


DEVELOP
class Agent():
def __init_(self):
def set_up(self):
def query(self, ... ):
local_agent = Agent( ... )
DEPLOY
remote_agent =
agent_engines.create(
local_agent,
requirements=[ ... ],
)
agent_engines.update( ... )
QUERY
remote_agent.query( ... )
# Streaming response
remote_agent.stream_query
( ... )
MANAGE
agent_engines.list()
remote_agent =
agent_engines.get(
agent_id
)
remote_agent.delete()
Create/Update API
Unary/Streaming API
List/Get/Delete API
Agent Engine


TUNISIA AI TOUR 2026
( 50% of the total income will go charity to people in need )

Devfest Istanbul 2025 -Presentation + Demo

EXPLORER

DEVFEST ISTANBUL 2025 -PRESENTAT...

devfest-agent-demo / devfe...

- .vscode
- venv
- agent_demo.py
- mock_api.py
- README.md
- requirements.txt
- train_model.py
- vertex_model.pkl
- Media
  - GDG Tripoli logo.png
- Ouput
  - Agent output.png
  - outpout 1-json.png
- venv
- DEMO-Agent Take Away.pdf
- README.md


OUTLINE

TIMELINE

CS-SCRIPT - ACTIVE

agent_demo.py

devfest-agent-demo > devfest-agent-demo > agent_demo.py > ...

```python
def model_predict(model, features_dict):
    pred_prob = model.predict_proba(X)[0, 1]
    pred_label = int(model.predict(X)[0])
    return pred_label, float(pred_prob) if pr

def explain_with_shap(model, X):
    explainer = shap.Explainer(model)
    shap_values = explainer(X)
    return shap_values

def execute_action(pred_label, payload):
    if pred_label == 1:
```

mock_api.py

devfest-agent-demo > devfest-agent-demo > mock_api.py > ...

```python
# mock_api.py
from flask import Flask, jsonify
import random
import time

app = Flask(__name__)

@app.route("/data")
def data():
    value = random.uniform(10, 100)
    features = {
        "f1": value + random.normalvariate(0,
        "f2": random.normalvariate(0, 1),
```

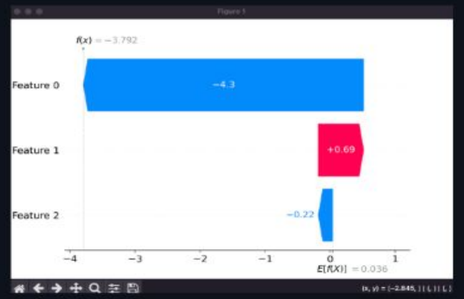
requirements.txt

devfest-agent-demo > devfest-ag

```
flask==2.3.3
scikit-learn==1.2
joblib==1.2.0
shap==0.42.1
matplotlib==3.7.1
pandas==2.1.2
numpy==1.26.2
requests==2.31.0
xgboost==1.7.6
```

Agent output.png

Ouput > Agent output.png




train_model.py

devfest-agent-demo > devfest-agent-demo > train_model.py > ...

```python
# train_model.py
import numpy as np
import pandas as pd
from sklearn.model_selection import train_tes
import xgboost as xgb
from sklearn.metrics import accuracy_score
import joblib

def make_sample_dataset(n=2000, seed=42):
    rng = np.random.RandomState(seed)
    f1 = rng.normal(loc=50, scale=15, size=n)
    f2 = rng.normal(loc=0, scale=1, size=n)
    f3 = rng.randint(0, 10, size=n)
```

PROBLEMS DEBUG CONSOLE TERMINAL PORTS

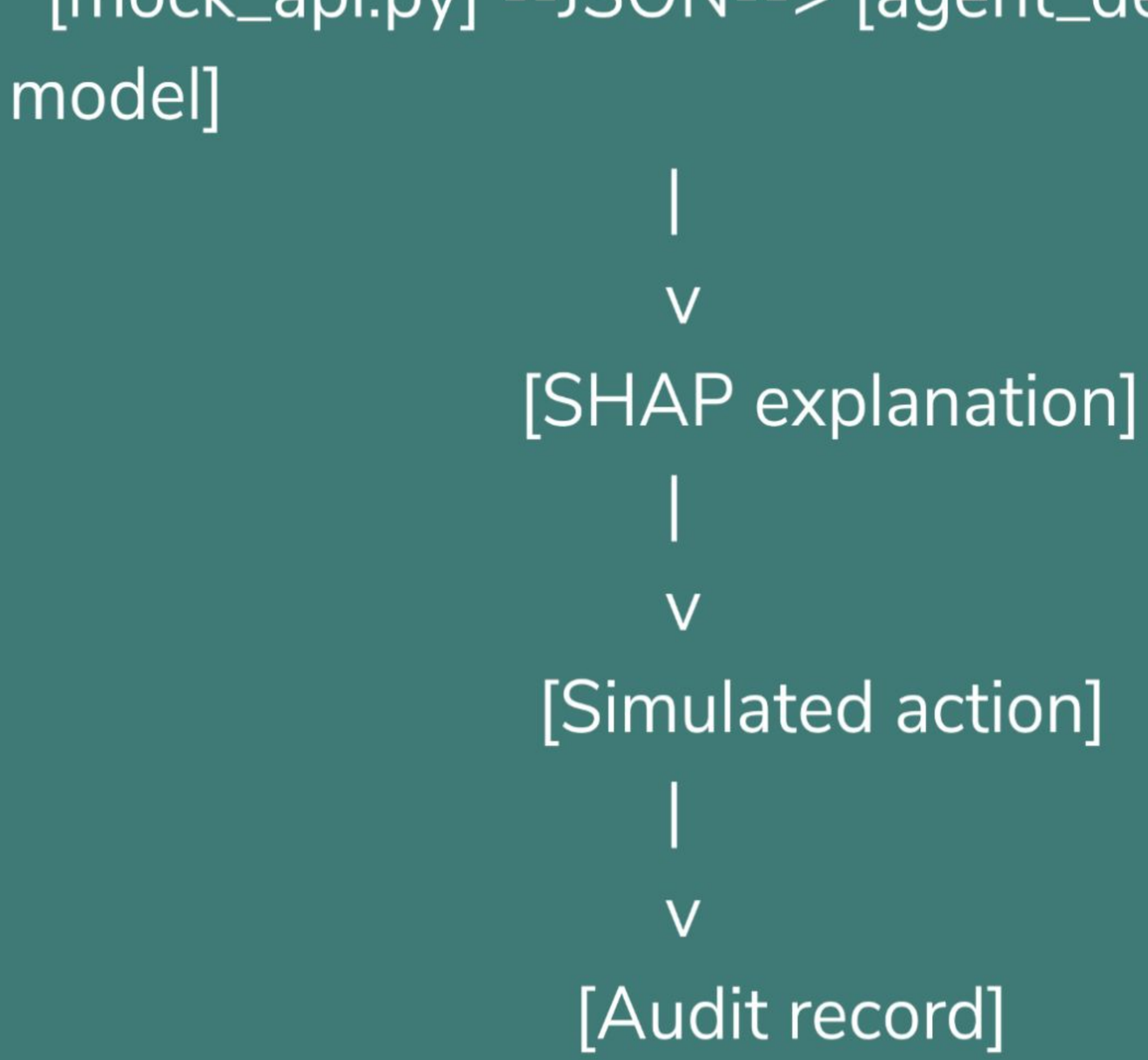
OUTPUT

TERMINAL zsh

```
Please update your Node.js version or visit https://nodejs
.org/ for additional instructions.
→  Devfest Istanbul 2025 -Presentation + Demo git:(main) ✗
```

main* 0 0 Cloud Code - Sign in

Whole Image 1588x1136 111.75KB Go Live





**Flow diagram of The demo**

# How this Enhances the Demo


**When We switch to Vertex AI:**

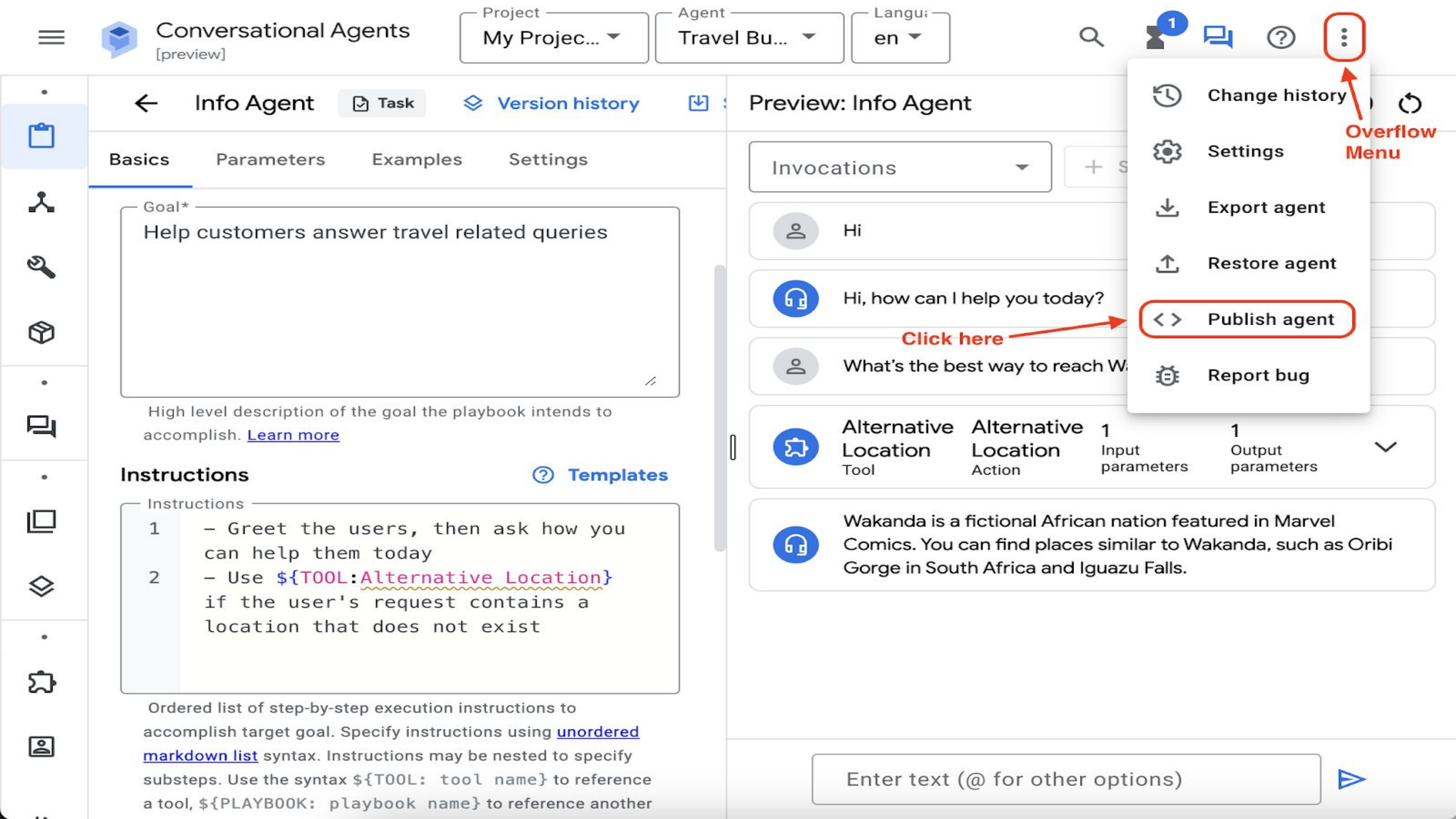
**agent_demo.py would call the Vertex AI endpoint for predictions**

```
from google.cloud import aiplatform

client = aiplatform.gapic.PredictionServiceClient()
endpoint = "projects/PROJECT_ID/locations/us-central1/endpoints/ENDPOINT_ID"

response = client.predict(
   endpoint=endpoint,
   instances=[features_dict]
)
```

**SHAP explanation can come from Vertex AI Explainable Predictions instead of running locally**

**The agent can handle real production-scale data instead of mock API**


https://codelabs.developers.google.com/devsite/codelabs/building-ai-agents-vertexai


Conversational Agents
[preview]
Project
My Projec...
Agent
Travel Bu...
Langu
en
Info Agent
Task
Version history
Basics
Parameters
Examples
Settings
Goal*
Help customers answer travel related queries
High level description of the goal the playbook intends to accomplish. Learn more
Instructions
Templates
Instructions
1 – Greet the users, then ask how you can help them today
2 – Use ${TOOL:Alternative Location} if the user's request contains a location that does not exist
Ordered list of step-by-step execution instructions to accomplish target goal. Specify instructions using unordered markdown list syntax. Instructions may be nested to specify substeps. Use the syntax ${TOOL: tool name} to reference a tool, ${PLAYBOOK: playbook name} to reference another
Preview: Info Agent
Invocations
Hi
Hi, how can I help you today?
Click here
What's the best way to reach W
Alternative Location
Tool
Alternative Location
Action
1
Input parameters
1
Output parameters
Wakanda is a fictional African nation featured in Marvel Comics. You can find places similar to Wakanda, such as Oribi Gorge in South Africa and Iguazu Falls.
Enter text (@ for other options)
Change history
Settings
Export agent
Restore agent
Publish agent
Report bug
Overflow Menu

```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to
Use `gcloud config set project [PROJECT_ID]` to change to a different project.
acloudpotato@cloudshell:~ $ ls
README-cloudshell.txt  sample-flask-app
acloudpotato@cloudshell:~ $ cd sample-flask-app/
acloudpotato@cloudshell:~/sample-flask-app $ ls
app.py
acloudpotato@cloudshell:~/sample-flask-app $ python app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 112-493-645
```

Click here to open web preciew as our application is running over port 5000 we need to change port to 5000

Flask app running in default port 5000

# Welcome to Travel Buddy!

Start chatting with our AI Travel buddy, in the bottom right corner.


Travel Buddy

Hi

Hi, how can I help you today?

What's the best way to reach Wakanda?

Wakanda is a fictional African nation featured in Marvel Comics. You can find places similar to Wakanda, such as Oribi Gorge in South Africa and Iguazu Falls.

Ask something...

# Resources

Vertex AI documentation resources :
https://cloud.google.com/vertex-ai/docs

SHAP library usage guides:
https://shap.readthedocs.io/en/latest/

Kubeflow Pipelines tutorials:
https://www.kubeflow.org/docs/

GitHub repository for this Presentation+Demo :
https://github.com/MortadhaMannai

# " Thank You For Your Attention! "