Problem - many AI models (like those from OpenAI), where inference happens in the cloud: your data gets sent to remote servers for processing, so you don't control exactly where the LLM calls go or which servers handle them.

With most AI models today (OpenAI, etc.):

👉 **Your data goes to someone else's cloud**
👉 **You don't know which server processed it**
👉 **You don't fully control data location or flow**

Before (traditional cloud LLMs)

Think of it like this:

🧑‍💻 You → 🌐 Internet → 🏢 OpenAI/Google server → 🌐 → You

- Data leaves your system
- You don't know which machine handled it
- You must *trust* the provider

After (Google's new approach)

Google's latest approach allows:

🧑‍💻 You → 🖥️ **Your own cloud / VPC / region** → You

- The **model runs where your data already is**
- Data **never leaves your boundary**
- You know **exactly which server handled the request**

Google is allowing companies to run Gemini (Google's best AI models) inside their own data centers instead of sending data to Google's cloud.

Google allows you to run AI models where your data already lives, instead of sending your data to external AI servers.

“Google Cloud allows us to create a Virtual Private Cloud, and using Vertex AI we can deploy AI models to run inside that private environment.Vertex AI provides private model endpoints and APIs that our applications use to send prompts and receive responses.
This gives us control over where the model runs by selecting the region, ensures data stays within our network, and allows us to monitor and audit all inference calls.”

Google Cloud lets us create a Virtual Private Cloud, which is our isolated private network. Using Vertex AI, we can deploy AI models in a specific region and expose them through **private endpoints** that are only accessible inside that VPC.

Our applications call these private Vertex AI APIs to send prompts and receive responses, and the traffic never goes over the public internet.

This gives us full control over **where the model runs**, ensures **data stays within our network and region**, and allows us to **log, monitor, and audit every inference call** using Google’s security and logging tools.”

While the model is thinking:

- Data is kept in **encrypted memory**
- Even Google cannot see it
- This uses **Confidential Computing**

Simple Example: Internal Company Chatbot

Scenario

Your company wants an **internal chatbot** that answers questions from **HR documents**.
The data is **confidential**, so it must **not go to public AI servers**.

Step-by-step (VERY simple)

Step 1: Private setup

- Company creates a **VPC** in Google Cloud
- Company deploys a **model using Vertex AI**
- Vertex AI gives a **private API endpoint**

Step 2: Employee asks a question

Employee types:

> “How many leave days do I have?”

Step 3: What happens behind the scenes

```
Employee Web App
   ↓ (inside private network)
Vertex AI Private Endpoint
   ↓
AI Model (runs securely)
   ↓
Vertex AI
   ↓
Employee Web App
```

Step 4: Security (in plain English)

- The question:
    - Does **not** go to OpenAI
    - Does **not** go to the public internet
- The AI:
    - Runs in the **company's cloud**
    - Processes data securely
- Only the **final answer** comes back

Step 5: Answer shown to employee

> "You have 24 leave days remaining this year."

One-line explanation you can say

> **"Our app talks to a private Vertex AI endpoint, the model runs securely inside our cloud, and the data never leaves our environment."**