

Google AI Leader Certification

08 September 2025 09:51

⌚ The Five Layers of Generative AI

- **1. Infrastructure:** This is the **foundation** of generative AI. It includes the physical hardware (servers, GPUs, TPUs) and software needed to store and run AI models and training data. 🏢
- **2. Models:** The "brain" of an agent. These are **complex algorithms** trained on huge amounts of data to learn patterns. They are what allows the AI to generate new content, translate languages, and answer questions. 💡
- **3. Platform:** This layer sits between agents and models. It provides the **tools and infrastructure** for AI development, such as APIs, data management, and model deployment tools. It helps simplify the complexities of the underlying infrastructure. 🔑
- **4. Agents:** These are pieces of software that use models and tools to **achieve goals autonomously**. They can analyze situations, use multiple tools, and handle multi-step tasks without constant human input. 🤖
- **5. Gen AI-Powered Applications:** This is the **user-facing layer**, or the frontend. It's what allows users to interact with and leverage the capabilities of generative AI. Examples include apps like Gemini and NotebookLM. ☐

Feature	RAG	Fine-tuning	Grounding
Definition	Augments LLMs by retrieving relevant information from external knowledge bases and adding it to the prompt.	Further trains a pre-trained model on a new dataset to adapt it to a specific task or domain.	Connects an AI model's output to verifiable sources of information.
Process	Retrieve relevant information. → Add it to the prompt. → Generate a response.	Select a pre-trained model. → Gather data. → Train the model. → Evaluate and refine.	Provide access to data sources. → Use RAG or fine-tuning to connect the output.
Data sources	External knowledge bases (databases, documents, internet).	Task- or domain-specific datasets.	External knowledge bases or specific datasets.
Relationship to grounding	A specific technique for achieving grounding.	Improves a model's ability to be grounded in specific domains.	The overarching goal, achieved through techniques like RAG and fine-tuning.

⌚ Why These Layers Matter

- Understanding these layers is **vital for leaders** to guide their teams in adopting and using generative AI effectively. 🏛️
- By understanding this interconnected system, leaders can better navigate the landscape and harness the power of AI for their own projects. 🎯

👤 Conversational Agents

Primary Goal	To understand and respond to user queries in a natural, human-like way.
Main Function	Understanding meaning and intent behind what you say.
Input Type	Typed or spoken messages from a user.
Tool Usage	Calls tools to gather information or perform actions related to a user's request, like searching the web.
Output Type	A relevant and natural-sounding response.
Example	A chatbot that answers your questions about a product.

⚙️ Workflow Agents

To automate and streamline tasks or complex processes.
Executing a series of pre-defined steps to complete a task.
A defined task, trigger (e.g., submitting a form, a scheduled event), or an online order.
Executes a sequence of actions based on a defined workflow, like transferring data or sending notifications.
A result or output from the completed task, such as a report, a confirmation message, or a data file.
An automated system that processes an order and sends a confirmation email.

AI agents are a significant leap beyond standalone models. They are designed to observe, act, and achieve goals by combining two key elements: a reasoning loop and tools. 🎯

What Distinguishes AI Agents?

- **Reasoning Loop:** This is the agent's "thinking process." It's a continuous, iterative cycle that enables agents to analyze a situation, plan actions, and adapt based on outcomes. The loop consists of four key steps:

1. **Observes:** Gathers information from its environment. 
 2. **Interprets:** Processes the information to assess the current situation. 
 3. **Plans:** Plans a course of action to achieve its goal. 
 4. Acts: Executes the planned action. 
- This cycle continues until the goal is met. The complexity of the loop varies from simple rules to complex thought chains and machine learning algorithms.
- **Tools:** These are functionalities that allow the agent to interact with its environment. Tools enable agents to access data, use software applications, and even communicate with physical robots.  This capability allows agents to connect with real-world information and services, much like apps on a phone.

Why They Matter

By combining reasoning and tools, AI agents can go beyond a single task and solve more complex problems that traditional large language models (LLMs) cannot.

- **Solving Complex Problems:** Agents can analyze information, gather data using tools, and make informed decisions with minimal human intervention. 
- **Managing Multi-Step Tasks:** Agents excel at complex workflows, such as conducting in-depth research, troubleshooting code, and automating tasks across multiple systems. 
- **Producing Diverse Results:** They can move beyond simple text generation to produce results that involve actions in the real world. 

The Challenge for Cymbal Aerospace

- **Manual and tedious process:** Their engineers had a hard time getting critical information like *material composition* and *dimensions* from complex 3D CAD models. 
- **Inefficient workflow:** This slow, manual process was prone to errors, which hurt their ability to efficiently design, analyze, and make jet engine components. 

The Generative AI Solution

- Cymbal Aerospace built a custom generative AI solution on **Vertex AI** using a two-layered approach. 
- **Layer 1: AutoML:** They used AutoML to *streamline the training process* for a machine-learning model that could extract **key metadata** from the CAD files.
 - This meant they didn't need a lot of deep machine-learning expertise to build a powerful model. 
 - The custom AutoML model acted as an "intelligent interpreter," understanding the complex CAD files and finding critical information.
- **Layer 2: Gemini:** They integrated Gemini to transform the raw metadata from the AutoML model into **clear, concise summaries** in natural language. 
 - This allowed engineers to ask simple questions like, "*What is the tensile strength of the turbine blade?*", and get an immediate, understandable, and accurate answer.
 - The integration was done using MLOps tools like *Vertex AI Pipelines*.

Benefits of Using Vertex AI

- **Simplified Model Development:** Vertex AI's AutoML handles the complex parts of model training, making it easy to build a high-performing model. 
- **Customized Solutions:** The company could train a model that was *perfectly tailored* to their specific needs and data. 
- **Efficient Scaling:** Vertex AI provided the infrastructure to **deploy and scale** the model, letting them process huge numbers of CAD files quickly. 
- **Streamlined Deployment:** They could easily deploy their models using a *serverless infrastructure*, so they didn't have to worry about managing the hardware.

The Impact

- The generative AI solution led to major improvements for Cymbal Aerospace:
 - **Product development cycles accelerated by 15%.** 
 - **Design review times were reduced by 20%.**
 - **Manufacturing bottlenecks were eliminated.**
- This faster time-to-market helped them respond quickly to customer demands and release innovative products ahead of their competition. 

High-Performance Computing (HPC)

- HPC is crucial for training complex generative AI models, which require immense processing power. 
- It helps solve the challenge of long training times and high infrastructure costs that many companies face with their in-house servers. 

Specialized Processors

- **GPUs (Graphics Processing Units):** Originally designed for graphics, GPUs are now widely used in AI because they excel at **parallel processing**, or doing many things at once. 
- **TPUs (Tensor Processing Units):** These are Google's custom-designed chips that are specifically **optimized for AI tasks**. 

- A **hypercomputer** is a supercomputer made by connecting many individual computers (nodes) that contain these GPUs and TPUs with high-speed networks. This creates one massive computing unit for training and running demanding AI models. 🚀

💾 High-Performance Storage

- Generative AI models are data-hungry and need to analyze massive datasets (often petabytes in size). 💾
- **High-performance storage** provides the necessary speed, capacity, and reliability to store and access these enormous datasets efficiently.
- Google Cloud's storage infrastructure is optimized for AI workloads, offering high throughput and scalability. ☑️
- **Fast storage access** is vital to prevent slow read and write speeds from hindering the training process. 🌐

🌐 Networking

- **Fast and efficient communication** is essential for coordinating all the processors in an HPC cluster. 🔍
- Google's global fiber network provides high-bandwidth, low-latency connectivity, ensuring smooth data flow between different parts of the AI infrastructure. ☐

- The key reasons for running AI models locally on a device, rather than in the cloud, are **real-time responsiveness**, **increased data privacy**, and the ability to operate **without an internet connection**. 🌐
- For example, a drone needs to react instantly to obstacles, making local processing essential for safety and efficiency. 🛳

🤖 Google's Tools for On-Device AI

- **Lite Runtime (LiteRT)**: This is Google's platform that helps machine learning models run efficiently on devices like smartphones. 📱
 - It optimizes models for on-device use by addressing constraints like latency, privacy, and power consumption.
 - LiteRT supports a wide range of models from frameworks like TensorFlow, PyTorch, and JAX.
- **Gemini Nano**: This is Google's most efficient and compact AI model, specifically designed to run on-device.
 - It's a miniature version of the more powerful models that typically live in data centers. 🌐
 - This on-device approach offers several benefits:
 - **Privacy**: Your data stays on your device. 🔒
 - **Speed**: You get fast responses without sending data to the cloud. ⚡
 - **Offline Access**: It can work even without an internet connection. ☑️

⌚ Using Vertex AI for Edge Deployment

- Even when deploying on the edge, you can use **Vertex AI** to build, train, and refine your models.
- Once your model is ready, Vertex AI provides tools to streamline the deployment process:
 - **Convert models** to Lite Runtime (LiteRT) for optimal performance on edge devices. 🌐
 - **Package and deploy** your models and their dependencies into containers for various edge hardware. ☑️
 - **Manage and monitor** your edge deployments to track performance and gather insights over time. 📈

👤 Roles and Responsibilities

- It's crucial to have the right people and expertise for your AI project.
- The AI stack supports a variety of roles:
 - **Business Leaders**: They use pre-built generative AI solutions (like Gemini for Google Workspace) to improve daily operations and customer experiences. 🏢
 - **Developers**: They build and deploy custom AI agents and integrate AI capabilities into existing apps. They can use platforms like Vertex AI for tools like orchestration, grounding, and action. ☑️
 - **AI Practitioners**: They are responsible for customizing, deploying, and optimizing generative AI models. Their expertise includes scaling AI workloads, integrating models with other services (like BigQuery), and implementing responsible AI practices. 🤖

💰 Cost in Generative AI

- The cost of generative AI depends on the product, company, and specific use case. Having a realistic budget is important. 📈
- You pay for three main activities:
 1. **Training** the model.
 2. **Deploying** the model to an endpoint.
 3. **Using** the model for predictions.

\$ Pricing Models and Metrics

- There are different ways providers charge for using models:
 - **Usage-based**: You pay for the amount you use, measured in tokens or characters processed. This is common for APIs. 📊
 - **Free tiers**: Some providers offer limited free access for experimentation. ☑️
 - **Licensing fees**: One-time or recurring fees for using a model, especially for commercial use. 🌐

- **Subscription-based:** You pay a recurring fee for access, often with tiers based on usage limits.
- Common metrics for usage-based pricing include:
 - **Tokens:** A token is a piece of text, like a word or a part of a word. 💰
 - **Compute time:** The time taken to process your requests. 🚧
 - **Requests:** Being charged per request, regardless of complexity.
 - **Characters:** The number of characters processed. 💬

💡 Factors Affecting Cost

- Several factors can impact the cost of your project:
 - **Model Size and Complexity:** Larger and more capable models generally cost more.
 - **Context Window:** A larger context window (the amount of text the model can consider) can increase costs.
 - **Specialized Features:** Features like fine-tuning or embeddings may have separate pricing. 💰
 - **Deployment:** Where you deploy your model and application can affect costs based on compute time. 🌐
- Remember, a pre-built generative AI application is fast and cheap, while a custom solution takes more time and resources.
 - 💡 It's important to match your project's timeline to your requirements.

⌚ Making Decisions for Your Gen AI Project

- When choosing a generative AI solution, you must carefully consider and research several factors. 🎓
- A well-researched decision will help you pick the most **cost-effective solution** for your needs. 💰

⚖️ Comparing Different Models and Companies

- **Evaluate model capabilities:** Look at independent benchmarks and research papers to check the *quality and performance* of different models. 📈
- **Compare pricing:** Understand the pricing models (like usage-based) and metrics (like tokens or characters) of each provider to estimate your costs. 💰
- **Factor in extra costs:** Don't forget to account for potential additional expenses like data storage, API calls, and the cost of fine-tuning a model. 💳
- **Read the fine print:** Always pay close attention to things like *usage limits*, *data privacy policies*, and other terms of service. 📄

📋 Key Resources for Comparison

- **Provider websites:** The best place to start is by checking the pricing pages on different company websites. 🌐
- **Research papers and benchmarks:** Look for independent evaluations and comparisons of different models. 📈
- **Community forums:** Engaging with other users and experts can give you valuable insights and real-world experiences with different models and pricing. 💬

🔧 Planning for Maintenance

- Building a generative AI solution is a long-term journey, not a one-time project. It's important to plan for its **ongoing maintenance and evolution** from the very beginning. 🛠️
- **Google Cloud** offers a fully managed environment that lets developers focus on building and improving the AI agent, rather than worrying about managing the underlying infrastructure and maintenance. 💰
- This approach can help you get the most value from your investment and avoid costly issues in the long run. 💫

⌚ What is Grounding?

- **Grounding** is the ability of an AI model to connect its output to **verifiable and specific sources of information**. 🎓
- This is a critical aspect for building *trust and reliability* in AI systems. ✅
- You can do this without any coding at all, just by providing the AI with resources like your company's style guide within the prompt or by uploading relevant files. 📁

🌐 Retrieval-Augmented Generation (RAG)

- RAG is a powerful technique for grounding an AI model. It works in two main steps:
 1. **Retrieving relevant information:** The AI model first retrieves relevant information from a vast knowledge base, like a database or a set of documents. This is often done using advanced techniques like *semantic search*. 🌐
 2. **Generating output:** The model then uses this retrieved information to generate the final output. 💡

🆚 Model-Only vs. Model with RAG

- **Model Only (without RAG):**
 - The model provides output *without* any external knowledge. 🤔
 - The output might be outdated and is more likely to have **hallucinations** (false or misleading information). 💰
- **Model with RAG:**
 - The model first sends a query to a backend system to find the most relevant information and includes it in the prompt. 💡
 - It then provides output with the latest external knowledge, which makes the output *much less likely* to have hallucinations. 💫

💡 Key Takeaways of RAG

- **Improved accuracy and relevance:** By using external knowledge, RAG models can produce more accurate and informative outputs. ⚡
- **Improved explainability and transparency:** RAG can increase trust by showing you the specific sources it used to generate the output, so you can check claims for accuracy. 📄
- **Extended LLM capabilities:** RAG can constrain the output from an LLM to generate a response *only* based on the specific context you provided. 🔒
- You can start using RAG without any coding or database development with tools like **NotebookLM**. 📎

● **Gemini for Google Cloud: An Overview**

- **Gemini for Google Cloud** is a set of AI-powered collaborators that are embedded across many Google Cloud products. 🤝
- It comes with standard Google Cloud enterprise protections, and your prompts or responses are *not used to train models*. 🔒

🔗 **Key Gemini Services**

- **Gemini Cloud Assist:** Like having an AI expert on your team. It helps you design, manage, and optimize applications. 📈
 - It analyzes your cloud environment and resources to provide *actionable, personalized guidance*.
 - It assists with application lifecycle management, including *troubleshooting, cost optimization*, and implementing *security best practices*. 🔑
- **Gemini in BigQuery:** Makes **data analysis easier and more accessible**. 📊
 - It can help you write code, understand your data, and *generate insights automatically*, regardless of your SQL experience.
 - This leads to *faster and more efficient data exploration*. 🌐
- **Gemini Code Assist:** An AI pair programmer for developers. 🤖
 - It provides code suggestions, generates entire code blocks, and offers explanations.
 - It supports over 20 popular programming languages, code editors, and developer platforms.
- **Gemini in Colab Enterprise:** A feature for **interactive notebooks**. 📈
 - It uses AI to help you write Python code by suggesting code segments and generating code from your descriptions.
 - This helps to streamline data analysis and machine learning workflows.
- **Gemini in Databases:** Helps developers and database administrators **manage databases more effectively**. 🗃
 - It uses AI to simplify tasks, from building applications with natural language to managing a fleet of databases from a single interface.
- **Gemini in Looker:** An intelligent assistant for **data analysis and insights**. 📈
 - It helps you understand your data, create visualizations, and *generate reports*, making data exploration more intuitive.
- **Gemini in Security:** Helps security teams **detect, contain, and stop threats** from spreading. 🛡️
 - It provides *near-instant analysis* of security findings and potential attack paths.
 - It summarizes tactics used by threat actors, providing detailed and timely threat intelligence.

🛠️ **Agent Tooling: Providing an Agent with Skills and Knowledge**

- Agent tools are the resources that give an agent the skills, connections, and knowledge it needs to achieve its goals. ⚡
- They allow agents to access information, perform actions, and interact with a variety of systems. 🤖
- We can break down agent tools into four key categories:

🛠️ **The Four Types of Agent Tools**

1. **Extensions (APIs):** These tools act as a bridge between an agent and **external APIs** (application programming interfaces). 🚀
 - They provide a standardized way for an agent to use an API, no matter the API's specific design.
 - **Example:** An agent that books travel could use an extension to interact with a travel company's API. The extension handles the technical complexities, so the agent can focus on finding and booking the flight. ✈️
2. **Functions:** These are **specialized tools** within the agent's toolbox that represent specific actions the agent can perform. 📈
 - An agent's reasoning system will select the correct function based on the task at hand.
 - Functions can encapsulate complex logic, making them reusable and easy to manage.
 - **Example:** A "*calculate_price*" function would take flight details and passenger information as input and return the total cost. The agent can then call this function whenever it needs to calculate a price. 💰
3. **Data Stores:** These tools give agents access to **information**. This can include real-time data, historical data, or knowledge bases. 📄
 - Data stores make sure that an agent's responses are *accurate, relevant, and up-to-date*.
 - **Example:** An agent could use a data store to access real-time weather conditions, current stock prices, or a database of customer information. 📊
4. **Plugins:** Plugins **add new skills or integrations** to an agent's capabilities. 🏷️
 - They can connect an agent to specific services or allow it to interact with particular platforms.
 - **Example:** A plugin could let an agent interact with a calendar app to schedule appointments. Another might integrate with a payment gateway to process transactions. 📱

Agent Tooling Example: Meeting Location Planner

- **Scenario:** A user uploads a document with a list of potential venues or meeting invitation details. The agent's goal is to suggest the most **convenient meeting location** for all attendees. ☈
- **Tools Used:** The agent uses a combination of different tools to achieve its goal:
 - **Document AI API:** To understand the content of the uploaded document.
 - **Google Maps API:** To analyze locations.
 - **Custom logic:** To suggest the final location.

The Agent's Workflow

1. **Document Processing:** The agent first uses the **Document AI API** to process the uploaded document and **identify all the addresses** mentioned inside it. 📄
2. **Geocoding:** Next, it uses the **Google Maps Geocoding API** to convert the extracted addresses into **latitude and longitude coordinates**. This allows the agent to calculate distances and travel times. 🌎
3. **Location Analysis:** Finally, the agent uses a **custom logic** (implemented within a Cloud Function) to analyze the geocoded addresses. This helps it determine and suggest the most convenient location for all the attendees. ☈
- This example shows how an agent can combine different tools to turn unstructured data (a document) into a helpful and actionable recommendation. ♦♦

How RAG Works with Tools

- **Retrieval-Augmented Generation (RAG)** enhances a large language model (LLM) by grounding its responses in **external knowledge sources**. 📄
- This allows the model to access and use information beyond its training data, which leads to more **accurate, relevant, and up-to-date** responses. ♦♦

The RAG Process with Tools

1. **Retrieval:** The LLM uses **retrieval tools** to find relevant information from outside sources. ☈ These tools can include:
 - **Data stores:** Internal databases or other sources of structured and unstructured data. 📄
 - **Vector databases:** Store numerical representations of text (**embeddings**) and find information that is semantically similar to the user's query.
 - **Search engines:** Used to find relevant web pages, articles, or other online content. 🌐
 - **Knowledge graphs:** Structured databases that store information about entities and their relationships. ☈
2. **Augmentation:** The retrieved information is then **added to the original prompt** that is given to the LLM. 📄 This augmented prompt now has both the user's request and the relevant context.
3. **Generation:** The LLM processes the augmented prompt and generates a response. Because it has the external information, the response is more **accurate and contextually appropriate**. It can also **cite its sources**, which increases transparency and trustworthiness. ☑
4. **Iteration:** In some systems, the LLM can **iterate on the retrieval process**. If the first search isn't good enough, it can refine its query or use different tools to find better information. It can also ask the user for clarifying questions. ☑

Data Stores in AI Applications

- Data stores are a key component of the RAG process. They act as **knowledge bases** that your agent can draw upon. 📄
- AI Applications allow you to connect your agent to a variety of data store options:
 - **Websites:** To access public information or stay up-to-date with current events. 🌐
 - **Structured data:** Information organized in tables or JSON format, like product catalogs or customer databases. 📊
 - **Unstructured data:** Files like HTML, PDF, and DOCX. 📄

What is Vertex AI Search?

- **Vertex AI Search** helps you create a **Google-quality search experience** for your public websites and applications. 🌐
- It can index and search across a wide variety of data, including:
 - Structured data in BigQuery.
 - Unstructured documents in Google Cloud Storage. 📄
- It uses your own data as the foundation for its responses, which helps reduce "hallucinations" and ensures the information is **trustworthy**. ♦♦ This is how it implements a **Retrieval-Augmented Generation (RAG)** approach.

Types of Search Solutions

- **Document Search:** Optimized for searching across a large repository of **unstructured, text-heavy documents** (e.g., internal knowledge bases). 📄
- **Media Search:** Specialized for searching within **images, videos, and audio files** based on their content, metadata, or even spoken words. 📹
- **Healthcare Search:** Designed for the unique needs of the healthcare industry. It allows for searching across healthcare data while ensuring **regulatory compliance**. 💶
- **Search for Commerce:** Specifically for e-commerce. It's optimized for product discovery and handling complex product queries for a retail catalog. 📄

Extra Generative AI Features

- **Search Summaries:** Vertex AI Search can generate **concise and informative summaries** of search results. This saves

users time and effort by providing a quick overview of a document or a synthesis of key findings. 

- **Answers and Follow-ups:** You can add AI-generated answers to your search results. Users can ask questions in natural language and then follow up with more questions, creating a conversational experience. 

Enterprise-Grade Features

- **Built for Enterprise:** It offers granular access controls to ensure **data security**. 
- **Advanced Analytics:** Provides analytics to help you understand search trends and user behavior. 
- **Scalable Infrastructure:** Designed to handle large volumes of data and search requests. 
- **Easy Integration:** It's built to seamlessly integrate with existing enterprise systems through APIs and SDKs. 

Building Conversational Agents

- There are two primary ways to build conversational agents: **deterministic** and **generative**. The best approach is often a combination of both. 
- **Deterministic agents** are rule-based and have a very defined system to follow (e.g., if a user presses a certain number, go to a specific route). They are more rigid and typically require low to medium code. 
- **Generative agents** are based on new generative AI technology and use large language models (LLMs) to give a *real conversational feel* to a chatbot. They can decide what to do on their own based on a prompt and often require no or low code. 
- **Hybrid agents** in Google Cloud combine the strengths of both, offering a powerful solution that provides strict control while still leveraging generative AI's flexibility. 

Agent Assist

- **Agent Assist** is a tool that supports *live human agents* with real-time, in-the-moment help. 
- It uses AI and generative AI to:
 - Recommend responses to customers. 
 - Suggest relevant content from a knowledge base. 
 - Transcribe or translate calls in real time.
 - Summarize conversations to help agents resolve issues faster and with greater accuracy. 
- This is especially useful for training new agents and ensuring consistency across all agents.

Conversational Insights

- **Conversational Insights** analyzes conversational data from across the customer journey to provide **data-driven insights**. 
- It helps contact center leaders and managers:
 - Boost efficiency and improve agent performance.
 - Understand metrics like agent and caller sentiment. 
 - Identify call topics and automatically flag interesting customer interactions for review.
- With the **Generative FAQ** feature, you can see common customer questions and how they are being answered, which helps identify gaps in your FAQs.

Contact Center as a Service (CCaaS)

- CCaaS is a **complete, cloud-based contact center solution** that simplifies the complexities of managing a modern contact center. 
- It manages the infrastructure, integrates with CRMs, and offers **omnichannel support**, which provides a consistent customer experience across all channels (phone, text, email, etc.).
- CCaaS integrates seamlessly with other tools like Agent Assist, Conversational Agents, and Conversational Insights, creating a full customer engagement suite. 

What is Agentspace?

- **Agentspace** is a central platform for enterprise-ready expert agents that are designed for your **specific business needs**. 
- It uses AI agents and a unified search to **automate tasks** and find information across **all** your connected business systems. 
- Agentspace is designed to be a **secure way** for businesses to deploy AI agents using Google Cloud's infrastructure. 
- You can think of it as your **personal AI assistant for work**, designed to increase productivity for a variety of teams (marketing, sales, HR, software, and R&D). 

Agentspace vs. NotebookLM

Agentspace

- Purpose** Your comprehensive enterprise AI assistant that automates tasks and finds information across all business systems. 
- Knowledge** Draws from all your connected business systems, breaking down information silos. 
- Actions** Can automate tasks and orchestrate workflows across your connected systems. 

NotebookLM Enterprise

- A specialized AI tool for deep research, summarizing, and creating content based *only* on specific documents you upload. 
- Focuses *solely* on the sources you provide, like documents or web pages. 
- Primarily a research and knowledge assistant that helps you understand and create content. 

Integrati A central platform that can *connect to* NotebookLM
on Enterprise. 

A tool for focused analysis that can be *integrated as a data source* within Agentspace.