Google Cloud

The Agent Opportunity

Connecting generative models to the outside world

Keelin McDonell

Product Manager,

Vertex AI,
Google Cloud

But first.....

AI Solution

Contact Center AI | Risk AI | Healthcare Data Engine | Search for Retail, Media and Healthcare

Gemini Agents

Build your own generative AI-powered agents

Vertex AI

Presentation Focus

Vertex AI Agent Builder

OOTB and custom Agents | Search
Orchestration | Extensions | Connectors | Document Processors | Retrieval engines | Rankers | Grounding

Vertex AI Model Builder

Prompt | Serve | Tune | Distill | Eval | Notebooks I Training | Feature Store | Pipelines | Monitoring

Vertex AI Model Garden

Google | Open | Partner

Google Cloud Infrastructure (GPU/TPU) | Google Data Cloud

Vertex AI

Enterprise-ready generative AI for builders

Best models from Google and the industry

End-to-end model building platform with choice at every level

Develop and deploy agents faster, grounded in your enterprise truth

Built on a foundation of enterprise readiness

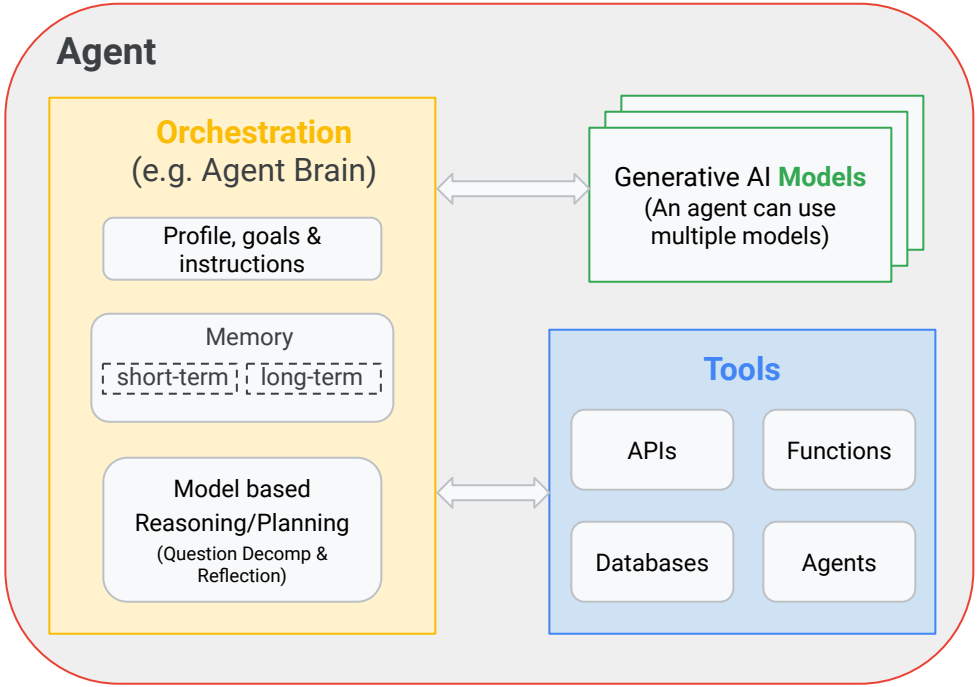
Vertex AI Enterprise-ready generative AI for builders

	Best models from Google & the industry	End-to-end model building platform with choice at every level	Develop and deploy agents faster, grounded in your enterprise truth	Built on a foundation of scale & enterprise readiness
Highlighted @ I/O	Gemini 1.5 Flash PaliGemma Gemma 2 Veo* Imagen 3.0	Batch API for Gemini API Context Caching Controlled Generation JSON Mode YAML, XML, others	RAG API Firebase Genkit (with Vertex AI Evaluation plug-in) Parallel Function Calling Grounding on Google Search Agent Builder API	Indemnification of outputs grounded with Google Search
Launched since NEXT	TimesFM Text Embedding Updates v4 of text Public Preview v2 of multilingual Public Preview	Embedding Tuning Public Preview Ray on Vertex AI GA PyTorch Model Co-Hosting on Vertex Endpoints Public Preview	LangChain on Vertex AI Public Preview	Dynamic Shared Quota

Now to Agents...

What is an “AI Agent”?

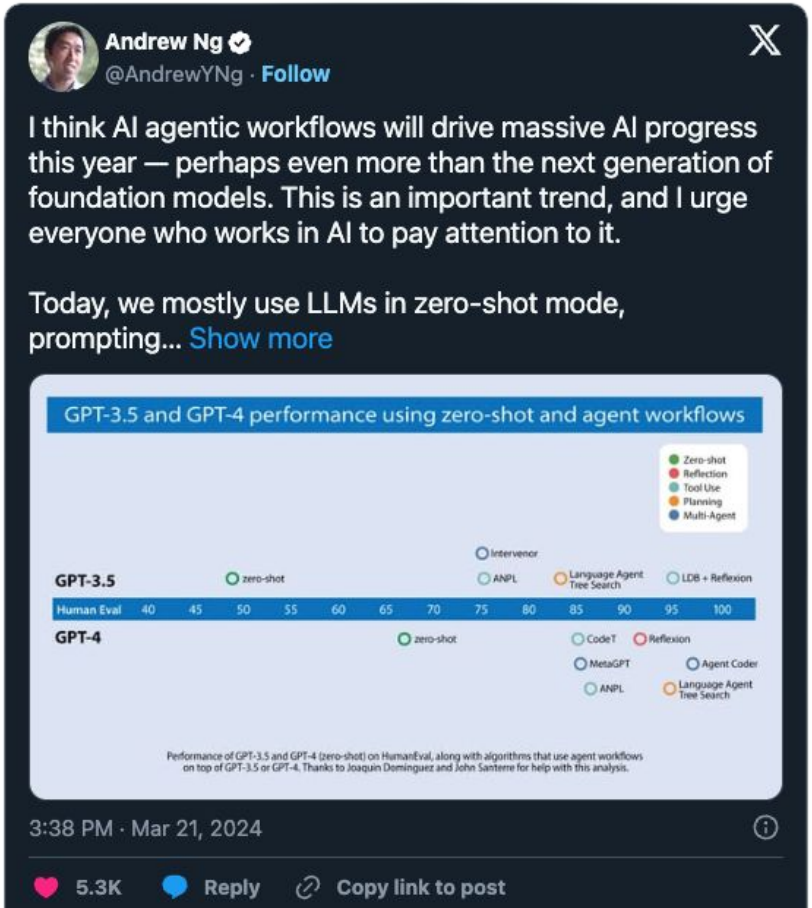
An application that **reasons** on how to best achieve a **goal** based on **inputs** and **tools** at its disposal



Key Components

- **Model:** Used to reason over goals, determine the plan and generate a response
- **Tools:** Fetch data, perform actions or transactions by calling other APIs or services
- **Orchestration:** Maintain memory and state (including the approach used to plan), tools, data provided/fetched, etc

Agents are how most people use LLMs already



Have you ever received a non-answer about outdated information?

```
As of my limited knowledge up to April 2023 [...] it's always best to check the latest rates from a reliable source, such as a currency converter or a bank.
```

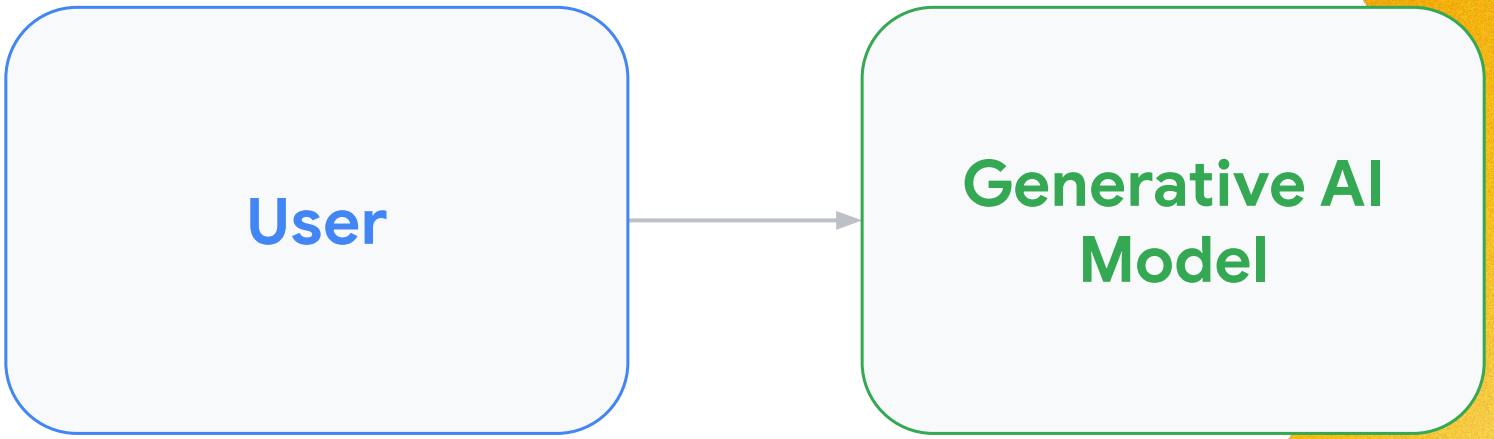
Have you ever tried to reference an external website, video, or API?

```
I am sorry, but as a language model,
I do not have the ability to access
external websites or videos.
Therefore, I cannot summarize the
article or the YouTube video you have
provided.
```

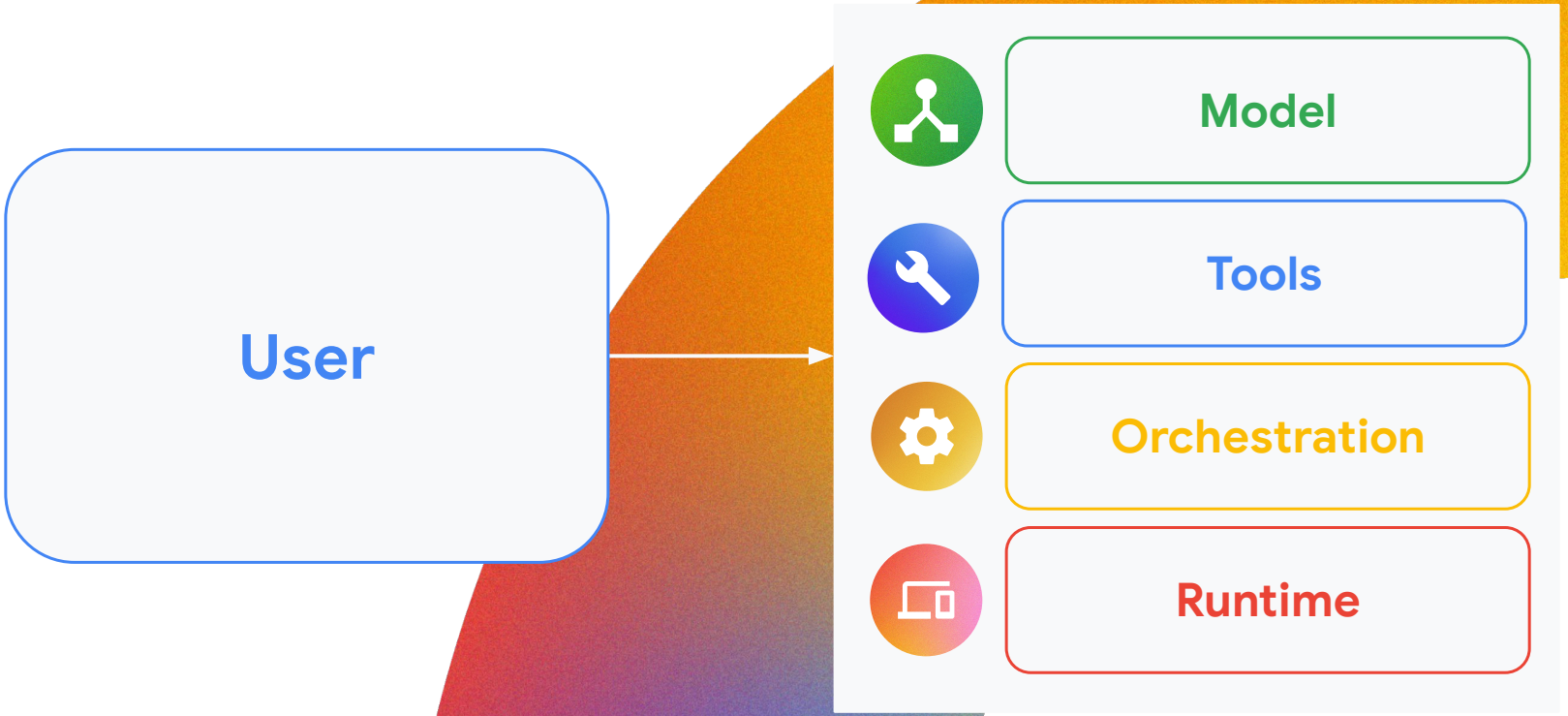
Have you ever tried to get consistent outputs?

```
```
{
"name": "alice",
"occupation": ["pets", "music"]
"address": false,
"email": "alice@example"
}
```
```

From models...



...to Agents



Choose your Model(s) in **Model Garden**

When you need a model for various tasks in your planned application

- Choose your flavor
- Choose your checkpoints
- Choose your fine-tuning
- Choose your eval

New!

Function Calling Extended Context

New!

Forced Function Calling

Function Calling UI Testing

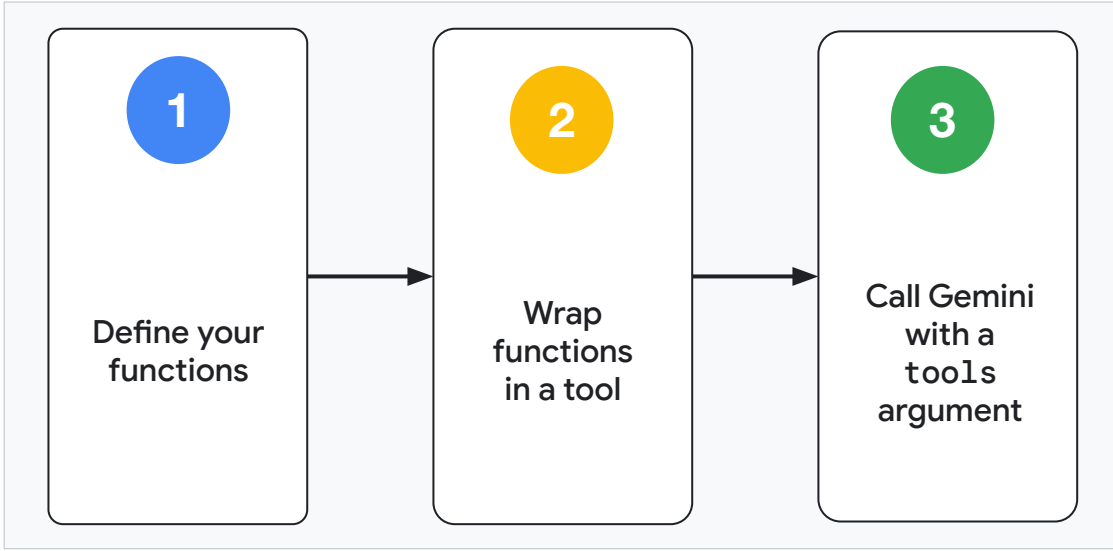
Function Calling Evaluation

Define your tools with **Function Calling**

Model

Tools

When you need Gemini to figure out the right API call to make based on a user query



Add your customized Agent Orchestration with **LangChain on Vertex AI**

Model
Tools
Orchestration

When you need orchestration for agent-like behavior, use LangChain templates to manage OSS, versioning, and more in Vertex AI!

```
class LangChainAgent:
    tools =
[StructuredTool.from_function(get_exchange_rate)]

    llm =
langchain_google_vertexai.chat_models.ChatVertexAI(
model_name="gemini-1.0-pro")

    self.agent_executor =
AgentExecutor(agent=agent, tools=tools,
verbose=True)
```

LangChain is a trademark of LangChain Inc. LangChain on Vertex is based on open-source LangChain version 1.1.13.

Deploy your agent to **Vertex AI**!

Model
Tools
Orchestration
Runtime

Use Vertex AI to productionize, deploy, and scale your agent with a simple API call

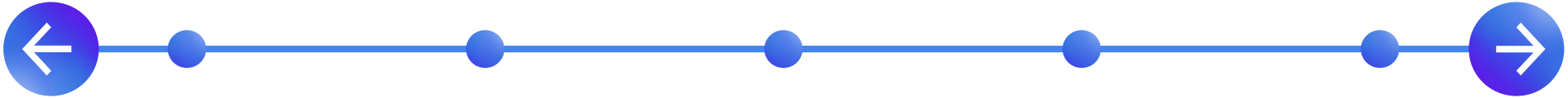
```
remote_app =
reasoning_engines.ReasoningEngine.create(
    LangChainAgent(),
    requirements=[
        "google-cloud-aiplatform",
        "langchain",
        "google-cloud-bigquery",
    ])

response = remote_app.query(input="What colors
does the Pixel 8 Pro come in?")
```

Function calling is about developer control and flexibility

Simple Tasks

Complex Workflows



Structured outputs

Parse unstructured data into structured fields

Information retrieval

Fetch information from APIs, databases, etc.

Intranet search

Help employees search and discover answers

Customer support

Help customers with questions, guidance, and orders

Autonomous workflows

Process database records and other batch data tasks

Use cases

Finance
Fetch real-time financial and currency exchange information

Business
Read and write to documents and spreadsheets in Google Drive

Travel
Fetch live flight and hotel information from travel systems

Customers
Search records in customer management systems

Databases
Perform real-time queries on datasets in BigQuery

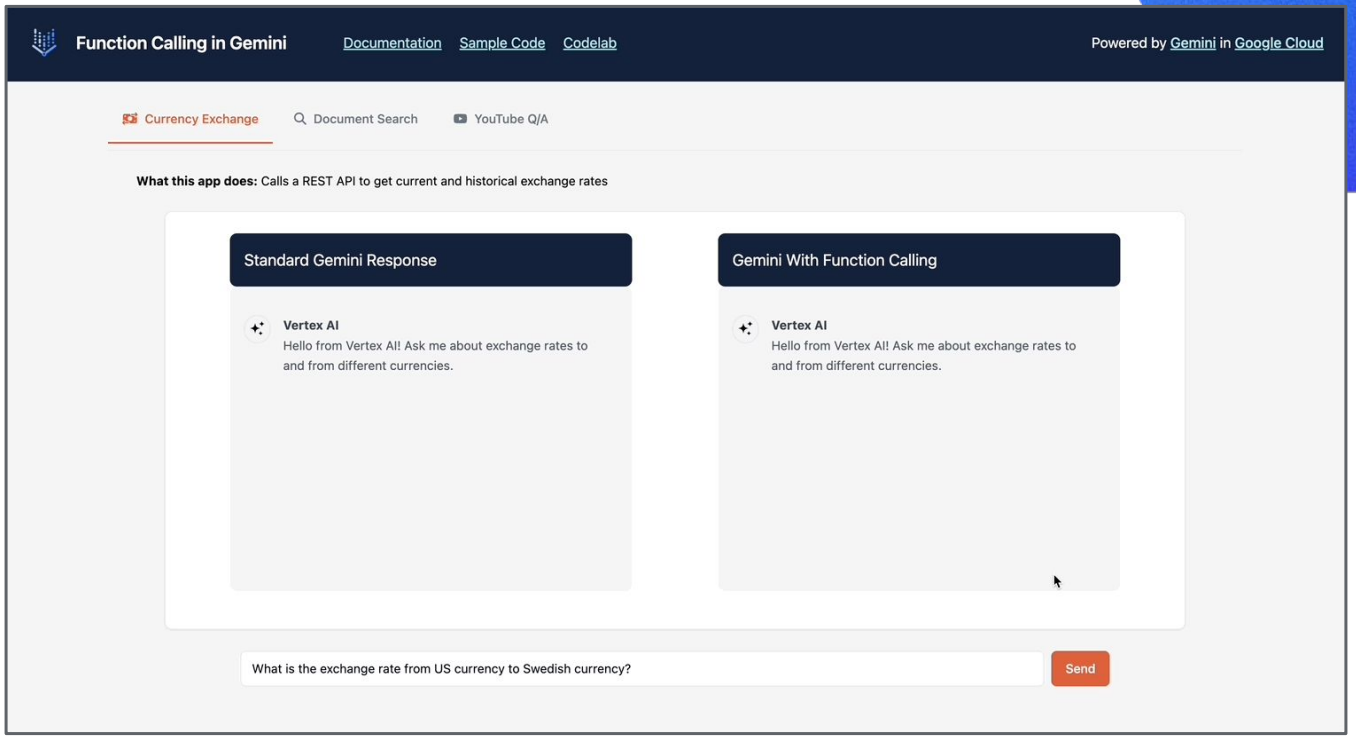
Documents
Search and summarize across thousands of documents

Support
Retrieve messages from customer support ticketing systems

Inventory
Make live queries to product inventory systems to check stock

Demo

Using Reasoning Engine to do Retrieval Augmented Generation (RAG)



Demo

Using Reasoning Engine to talk to Google Drive and Google Sheets

Invoice Processor

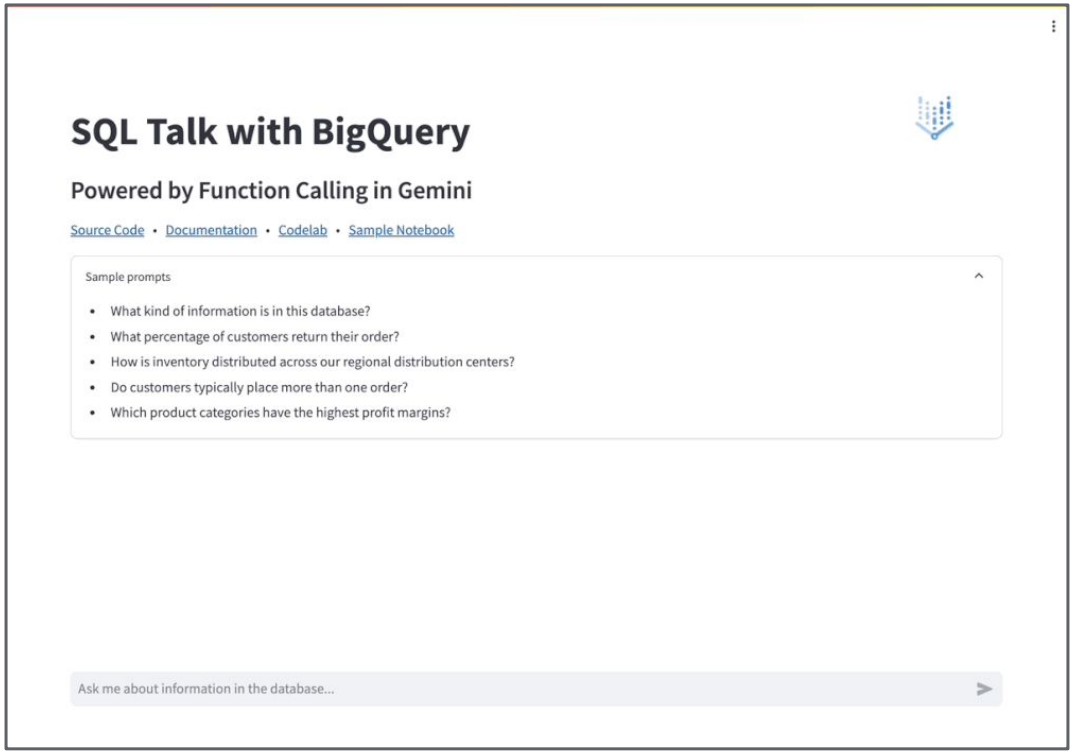
Powered by Gemini Function Calling, Reasoning Engine, and LangChain

List of all incoming invoices

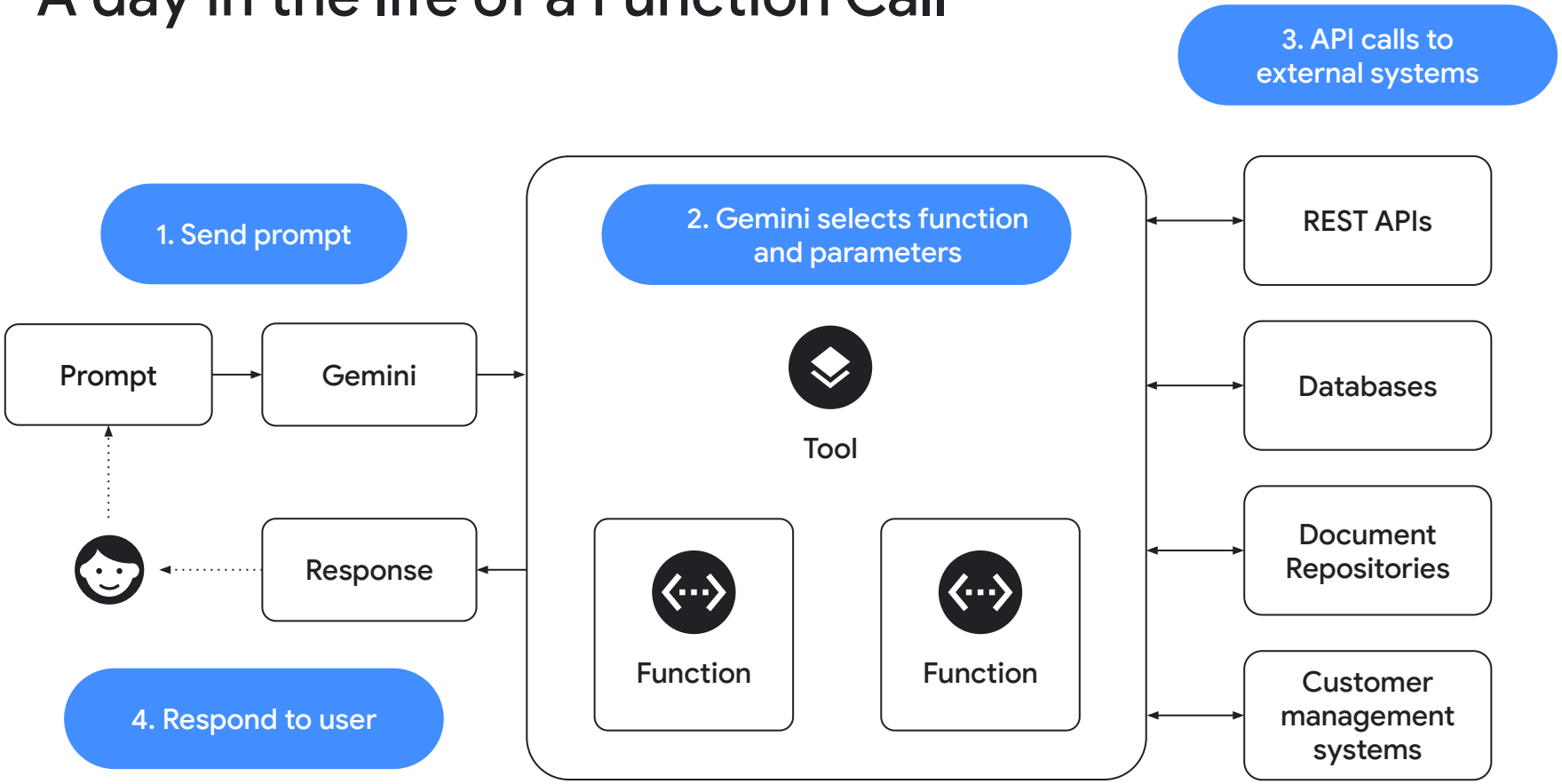
Invoice URL
https://storage.googleapis.com/gemini-fc-re-invoice-demo/invoices/ams_synthetic_file_1660294492573143.pdf
https://storage.googleapis.com/gemini-fc-re-invoice-demo/invoices/ams_synthetic_file_1660294503342853.pdf
https://storage.googleapis.com/gemini-fc-re-invoice-demo/invoices/ams_synthetic_file_1660295757263746.pdf
https://storage.googleapis.com/gemini-fc-re-invoice-demo/invoices/ams_synthetic_file_1660295767846476.pdf
https://storage.googleapis.com/gemini-fc-re-invoice-demo/invoices/ams_synthetic_file_1660296508641677.pdf

Demo

Using Reasoning Engine to talk with SQL databases



A day in the life of a Function Call



Using LangChain on Vertex AI

1

Define function(s)

```
def get_exchange_rate(
    from: str = "USD",
    to: str = "EUR"):

    "Retrieves the exchange rate."

    response = requests.get(
    "https://api.currency.app/",
      params={"from": from,
              "to": to})

    return response.json()
```

2

Use LangChain templates

```
app =
llm_extension.LangchainAgent(

    tools=[get_exchange_rate],

    model_kwargs={
      "temperature": 0.3,
      "top_p": 1,
      safety_settings: {...},

    }
)
```

3

Deploy to Vertex AI

```
remote_app =
reasoning_engines.ReasoningEngine.
create(
    LangChainAgent(),
    requirements=[
      "google-cloud-aiplatform",
      "langchain",
      "requests==2.*"])

remote_app.query(
  query="What's the exchange rate
  from US dollars to Swedish
  currency?")
```

Full control in development

Define functions, tools, parameters, and API calls and let Gemini handle the hard part of selecting an appropriate function and extracting parameters from prompts.

```
def get_directions(origin, destination):
    api_key = "YOUR_API_KEY"
    maps_client = Client(api_key)
    directions = maps_client.directions(...)
    return directions


def translate(text, target_language):
    client = translate_v2.Client()
    result = client.translate(text, ...)
    return result['translatedText']


def upload_to_gcs(file_path, bucket_name):
    client = storage.Client()
    bucket = client.get_bucket(...)
    blob = bucket.blob(...)
    blob.upload_from_filename(...)
```

Less boilerplate code

Reasoning Engine uses Function Calling in Gemini to invoke functions as tools without requiring the use of verbose prompt templates or manually piping strings between components.

```
# Define an agent in Reasoning Engine that uses
Gemini Function Calling and LangChain

app = llm_extension.LangchainAgent(
    tools=[get_exchange_rate],
    model_kwargs={
      "temperature": 0.3,
      "top_p": 1,
      safety_settings: {...},
    }
)

# Reuse Reasoning Engines in your app code

reasoning_engine =
llm_extension.ReasoningEngine("projects/PROJECT_ID
/locations/LOCATION/reasoningEngines/REASONING_ENG
INE_ID")
response = reasoning_engine.query(query=query)
```

Fast prototyping

Build faster to explore new ideas without waiting on a connector for a specific service or API to be released. Connect Gemini to any API directly with LangChain on Vertex AI.

```
def support_system(ticket_number):
    ...

def document_repository(query):
    ...

def code_search(git_repo):
    ...

def graph_database(statements):
    ...

def your_custom_service(params):
    ...
```

As a market researcher...

With deep knowledge of an industry but limited or no coding experience...

- **I want to:** Ask business questions in natural language and receive visualizations and analytical commentary.
- **So that:** I can accelerate the time to insight by quickly exploring a dataset without the need to become a coding expert or sending a request to our data processing department.

Table of Contents

1	144. Are you currently working from home, at your workplace, or both a majority of the time?
2	144. Are you currently working from home, at your workplace, or both a majority of the time?
3	144. Are you currently working from home, at your workplace, or both a majority of the time?
4	144. Are you currently working from home, at your workplace, or both a majority of the time?
5	144. Are you currently working from home, at your workplace, or both a majority of the time?
6	144. Are you currently working from home, at your workplace, or both a majority of the time?
7	168. In your opinion, what is closest to the right mix of working from home versus working in an office right now?
8	168. In your opinion, what is closest to the right mix of working from home versus working in an office right now?
9	168. In your opinion, what is closest to the right mix of working from home versus working in an office right now?
10	168. In your opinion, what is closest to the right mix of working from home versus working in an office right now?
11	168. In your opinion, what is closest to the right mix of working from home versus working in an office right now?
12	168. In your opinion, what is closest to the right mix of working from home versus working in an office right now?
13	379_NEW_W67. How likely, if at all, is it that you will keep your current work arrangement over the next few months?
14	379_NEW_W67. How likely, if at all, is it that you will keep your current work arrangement over the next few months?
15	379_NEW_W67. How likely, if at all, is it that you will keep your current work arrangement over the next few months?
16	379_NEW_W67. How likely, if at all, is it that you will keep your current work arrangement over the next few months?
17	379_NEW_W67. How likely, if at all, is it that you will keep your current work arrangement over the next few months?
18	379_NEW_W67. How likely, if at all, is it that you will keep your current work arrangement over the next few months?
19	380_NEW_W67. Has your employer set guidelines for how often you should work from the office or workplace?
20	380_NEW_W67. Has your employer set guidelines for how often you should work from the office or workplace?
21	380_NEW_W67. Has your employer set guidelines for how often you should work from the office or workplace?
22	380_NEW_W67. Has your employer set guidelines for how often you should work from the office or workplace?
23	380_NEW_W67. Has your employer set guidelines for how often you should work from the office or workplace?
24	380_NEW_W67. Has your employer set guidelines for how often you should work from the office or workplace?
25	70_NEW_W9. In the next 3-5 years, do you expect your work commute to change?

1. Tens of thousands of consumer interviews
2. Hundreds to thousands of survey questions
3. Dozens to hundreds of waves of data

100s to 1,000s of sheets

Index | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16

Ipsos consumer COVID-19 tracker W91

144. Are you currently working from home, at your workpla

	Age			Household Income				Region				PID			
	18-34	35-54	55+	Under $50K	$50K-<$100K	$100K+	$125K+	Northeast	Midwest	South	West	Republican	Democrat	Independents	Rural
	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
Base: Employed (unwtd)	**203**	**234**	**118**	**171**	**226**	**158**	**96**	**130**	**121**	**190**	**114**	**212**	**230**	**78**	**89**
Base: Employed (wtd)	**207**	**270**	**118**	**113**	**168**	**314**	**226**	**115**	**126**	**204**	**150**	**237**	**238**	**84**	**104**
Working from home only	29	74	36	25	31	83	66	21	23	41	53	55	60	18	24
	14%	27%	30%	22%	18%	26%	29%	18%	19%	20%	35%	23%	25%	22%	23%
		D*	D*			*	*	*	*	*	*	*		*	*
Working at my workplace only	119	122	59	65	98	137	91	66	74	106	54	134	100	43	65
	57%	45%	50%	58%	59%	44%	40%	58%	59%	52%	36%	57%	42%	51%	63%
		*	*	J	IJ	*	*	N*	N*	*	*	*		*	*
Working both from home and at my workplace	59	74	23	23	39	94	70	28	29	57	43	48	78	23	15
	29%	27%	20%	20%	23%	30%	31%	24%	23%	28%	29%	20%	33%	27%	14%
		*	*			*	*	*	*	*	*	*		*	*
Sigma	207	270	118	113	168	314	226	115	126	204	150	237	238	84	104
	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%

Field Dates: 2/21-2/22

Statistics:

Overlap formulae used

- Column Proportions:

  Columns Tested (5%): A/B/C,D/E/F,G/H/I/J,K/L/M/N,O/P/Q

  Minimum Base: 30 (**), Small Base: 100 (*)

- Column Means:

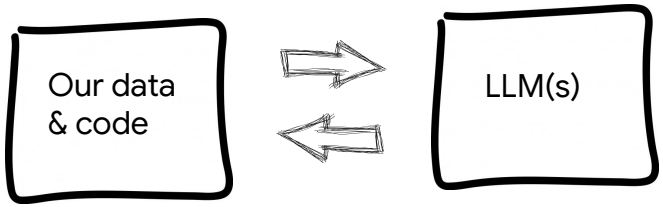
  Columns Tested (5%): A/B/C,D/E/F,G/H/I/J,K/L/M/N,O/P/Q

  Minimum Base: 30 (**), Small Base: 100 (*)

Table of contents

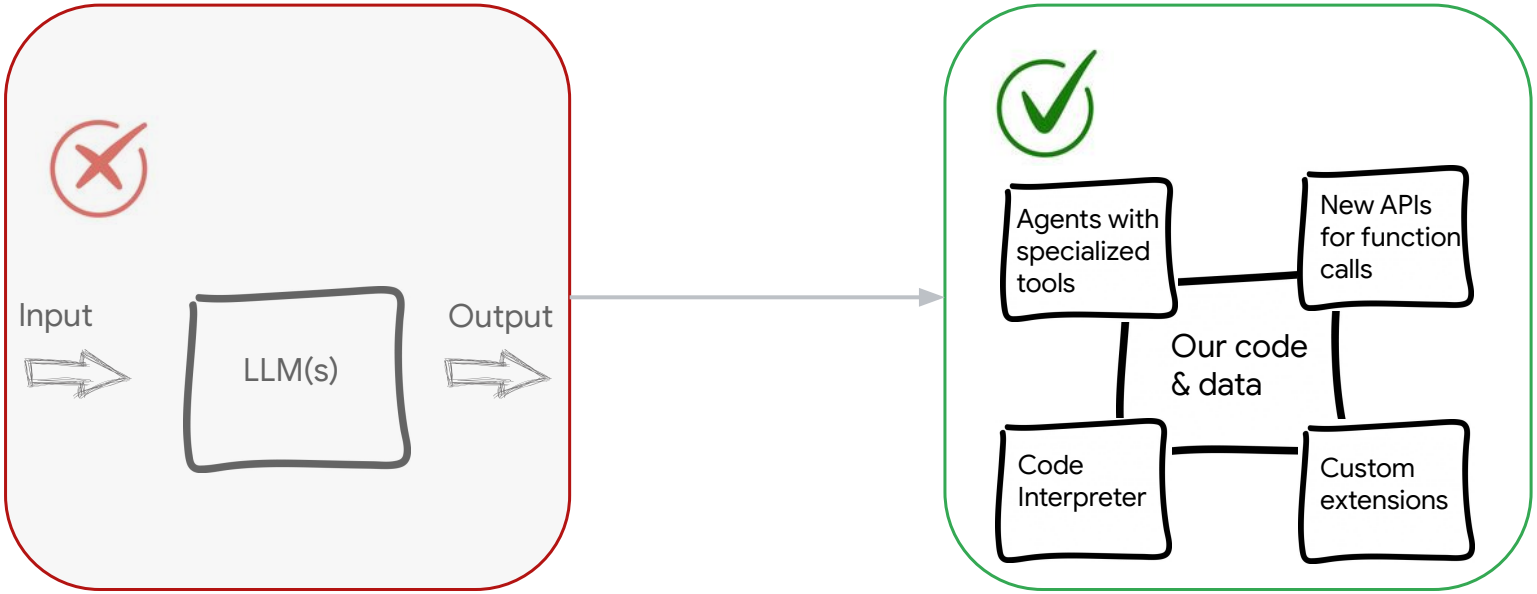
10s to 1,000s of columns

The black box / off-the-shelf approach did not work...



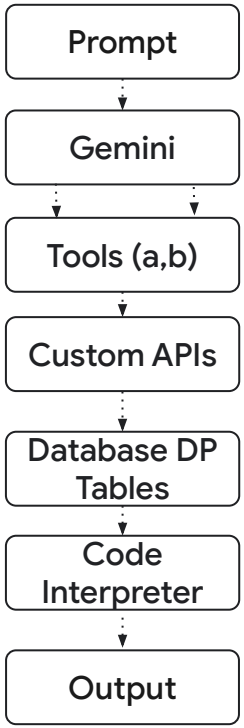
1. Low quality answers
2. Poor UX in the form of “prompt engineering”
3. Lack of differentiation & specialization

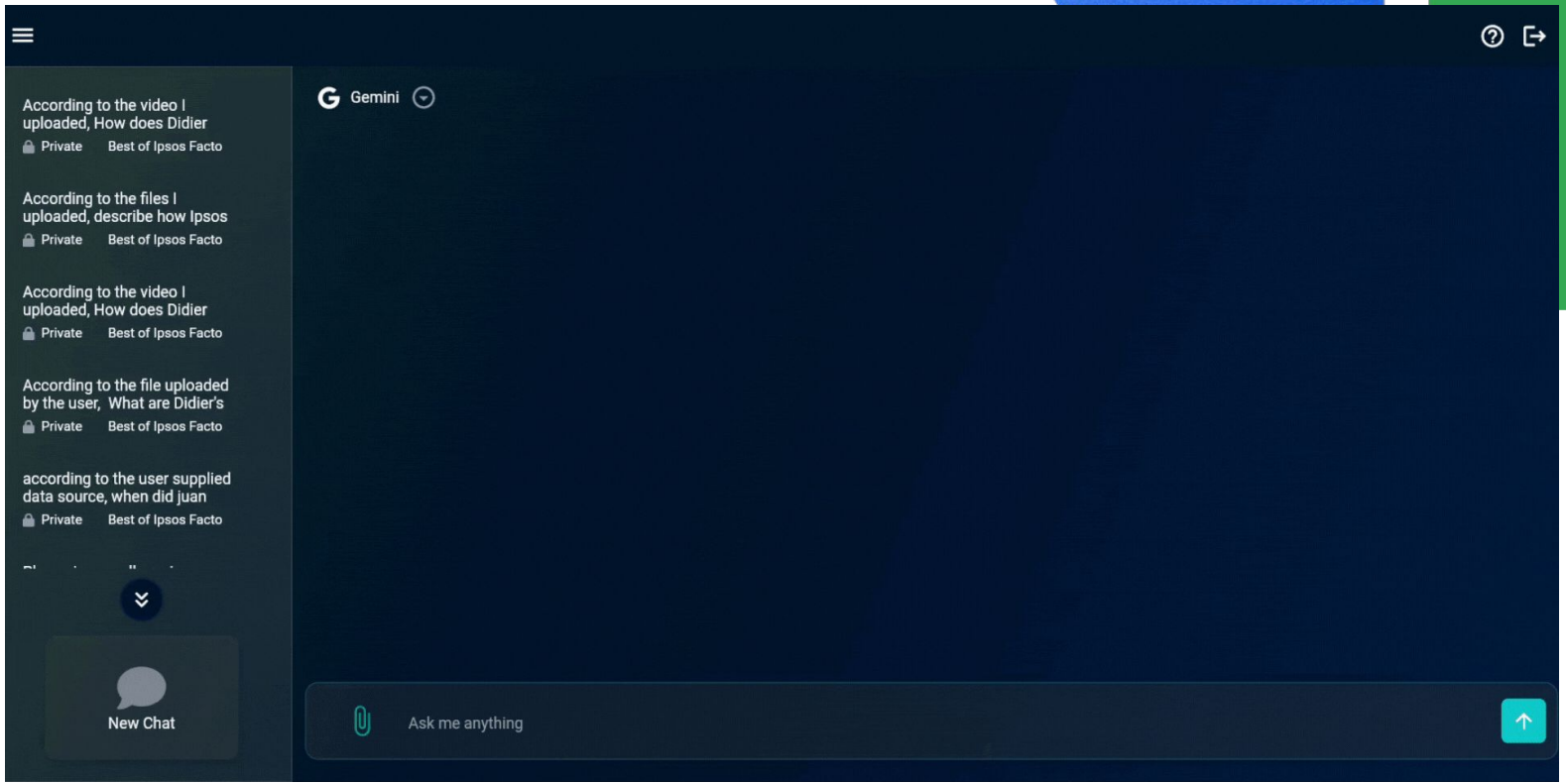
A purpose-built toolset...



Demo

Example: Are americans interested in spatial computing?





Lessons learned

1. The LLM only approach does not work for complex use cases – agents are key!

2. To drive performance and differentiation we need control at every layer of the GenAI stack through agents

3. Combine different solution designs, models, orchestration requirements and tools

Vertex AI Free Trial

Scan the code to start a free trial of Vertex AI

Join our Innovators Program

Google Cloud Innovators

The Innovators program gives developers and practitioners the latest updates, access to technologies and expertise, and exclusive benefits to build your skills on Google Cloud.

cloud.google.com/innovators

Google Cloud

Thank You