# Navigating the New Frontier Lessons from Building and Deploying Agentic Al Apps

April 3rd, 2025

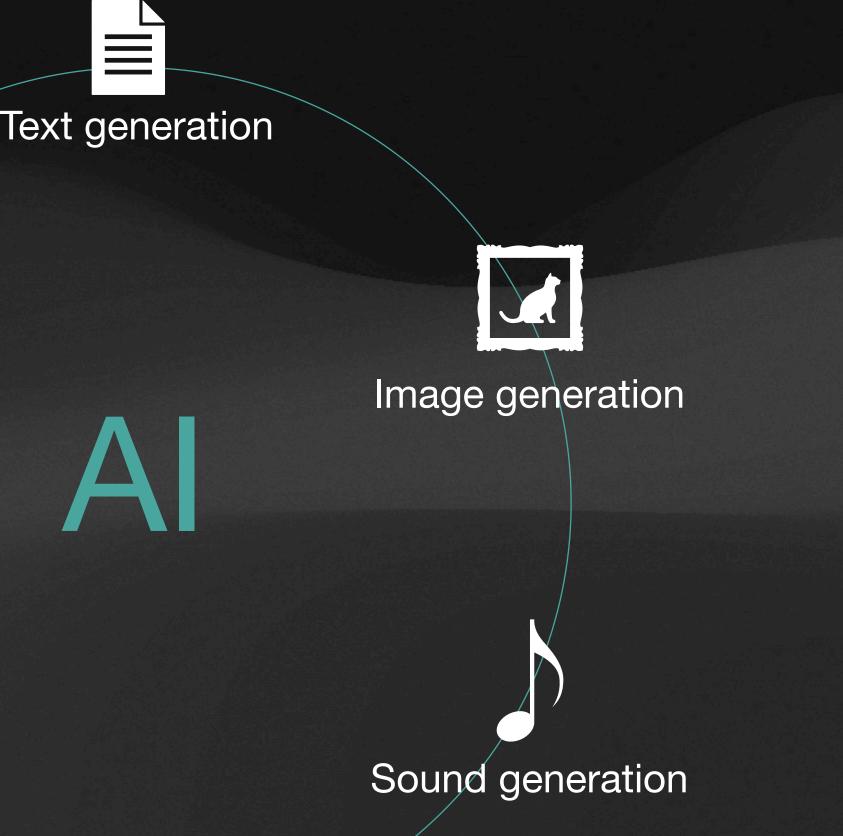
Juan Peredo



### Welcome to the renaissance of computing









### Interacting with the world via agents

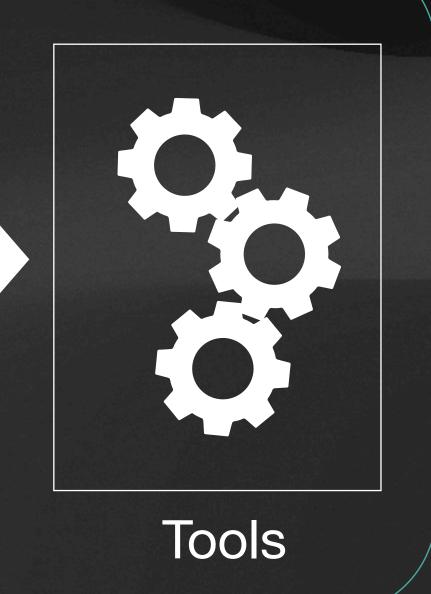
Uses

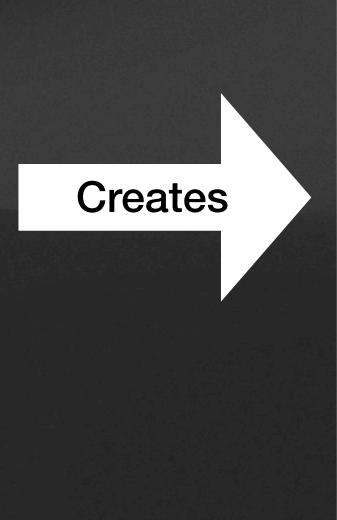






User

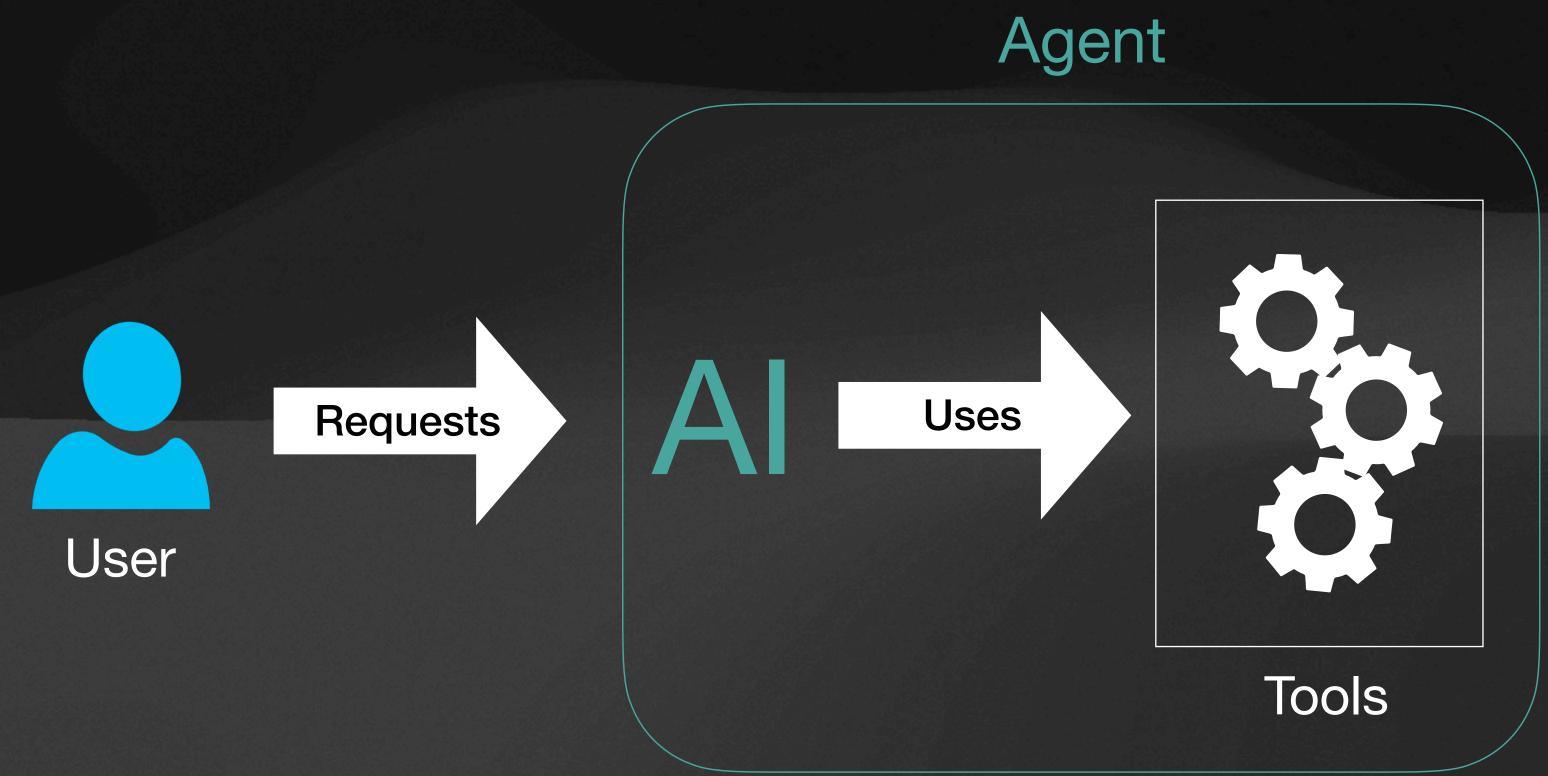




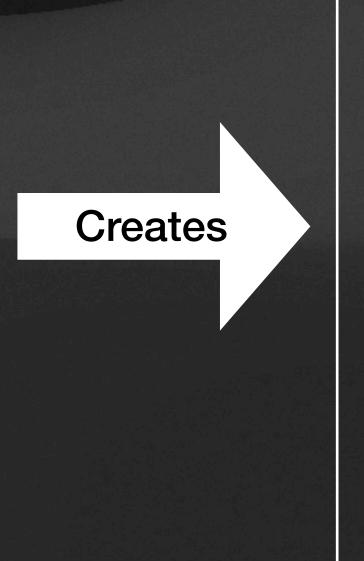
Translations Trip bookings Video tutorials Customer support Interactive stories And so much more!



### Well... At least trying to interact with the world ...



Al introduces a lot of new variables into a product creation journey



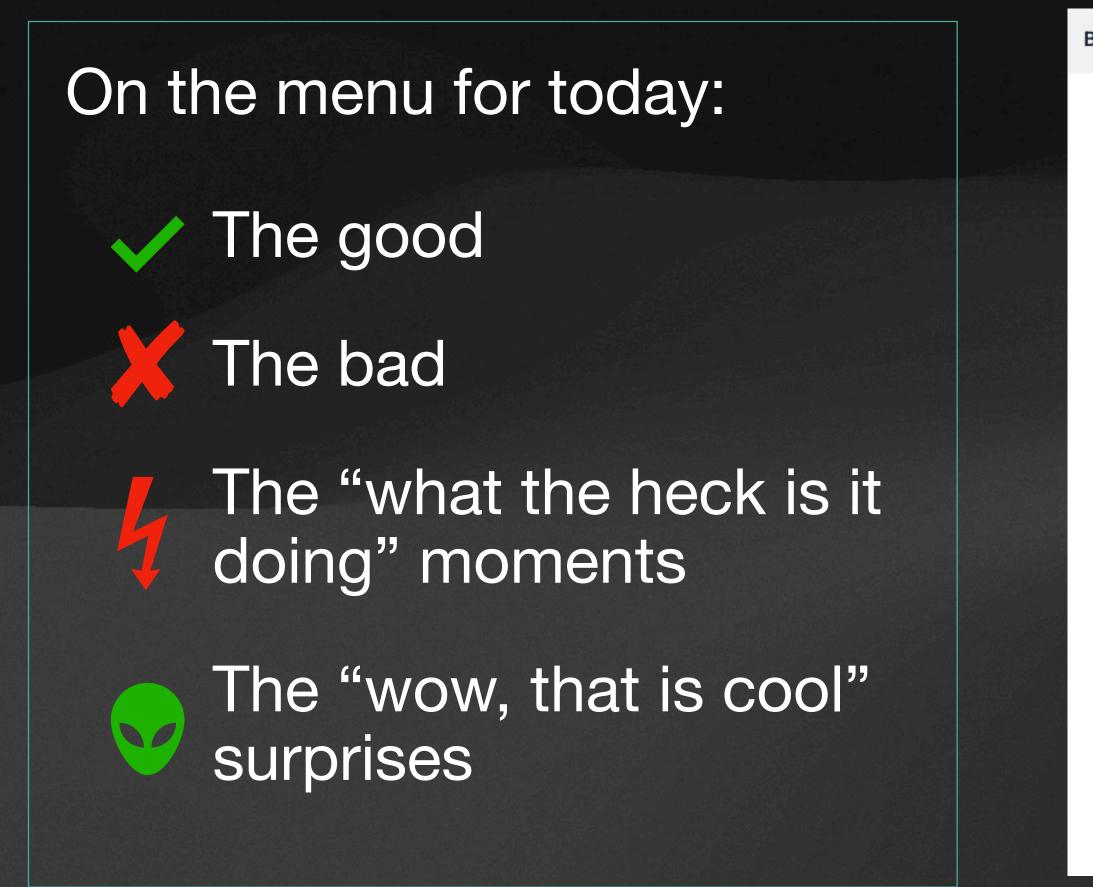
Hallucinations Weird errors **Budget overflows Incorrect** information Hard to test interactions And so much more!



## Juan Peredo - Your guide in this journey

- Founder / architect / consultant / developer & everything in between
- Linkedin: <u>linkedin.com/in/juanperedotech</u>
- Over 15 years of IT experience in companies like:
  - Bolbeck LLC
  - AWS
  - Strategy& (PWC)
  - Booz & Co

### A look into the trenches of App dev with agents



**Bolbeck Imagine** 

Sign up Pricina Sign in

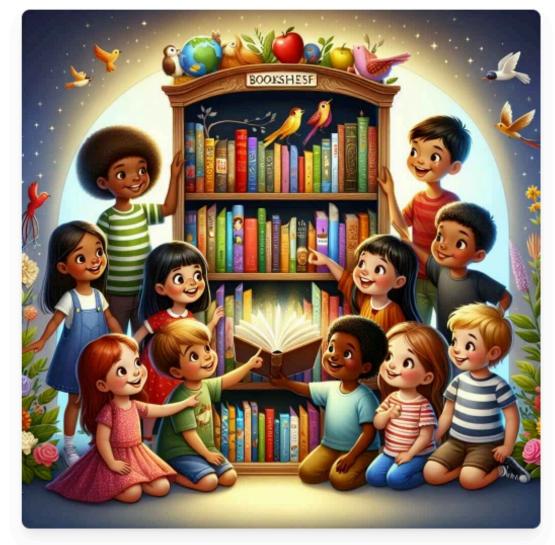
**Bring your children's** books to life

Uncover the magic of books with Bolbeck Imagine, the cutting-edge AI platform that enhances and simplifies the creation and distribution of children's books.

Make children worldwide fall in love with reading all over again!

Would you like to lend a hand in our testing and be the first to know when we launch?

Join our early adopters program!



Imagine all you could accomplish

### This is my journey. Your mileage may vary.



# Let's take a quick look at the app... It's demo time!

**Bolbeck Imagine** 

#### **Bring your children's** books to life

Uncover the magic of books with Bolbeck Imagine, the cutting-edge AI platform that enhances and simplifies the creation and distribution of children's books.

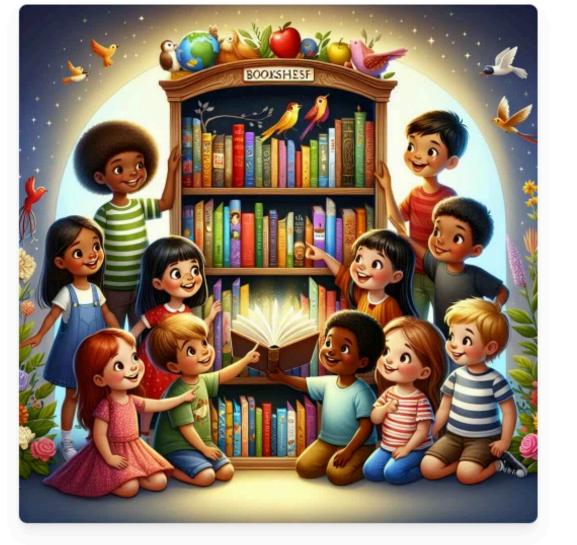
Make children worldwide fall in love with reading all over again!

Would you like to lend a hand in our testing and be the first to know when we launch? Join our early adopters program!

Imagine all you could accomplish

Pricing About Sign in Sign up





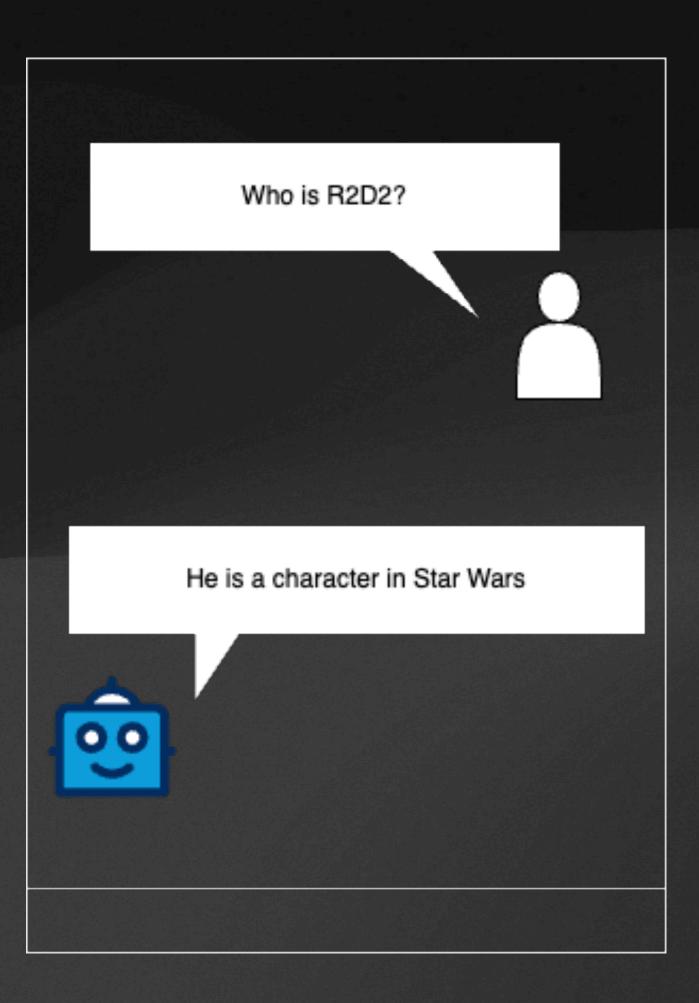
### Going back to basics

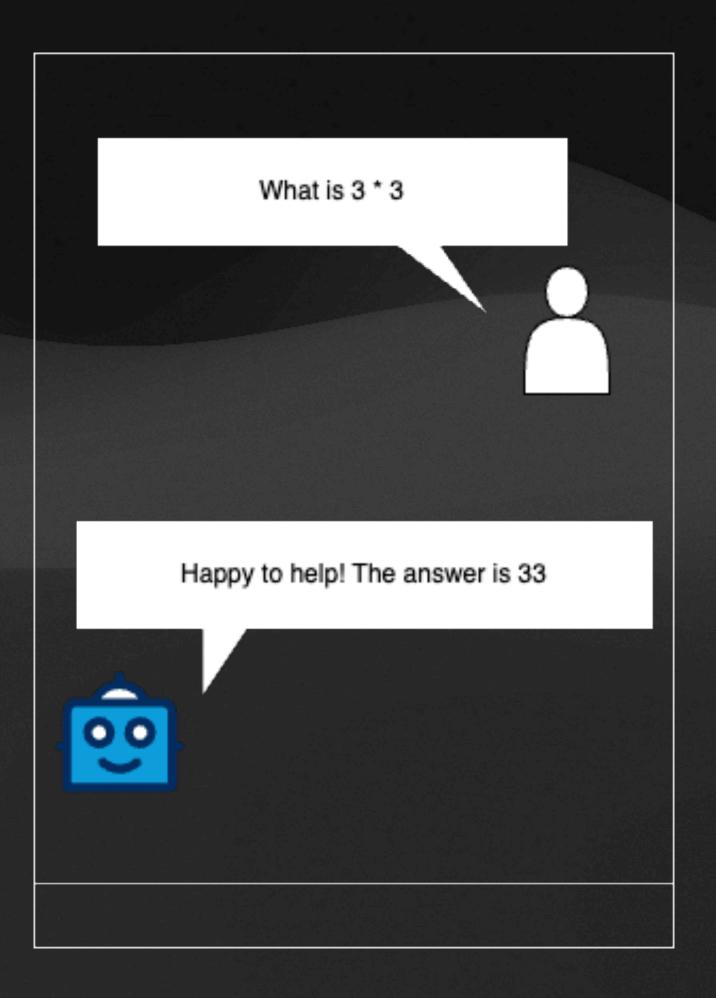
# How it all got started...

### Going back to basics



## Building an Al chatbot is easy





### However, validating and moderating the chatbot content is hard

## There are techniques to validate LLM output

### Technique

**Prompt Engineering** 

#### Description

Provide directions to LLM in the prompt

Guardrails

Specialized LLMs that classified answer as 'bad'

RAG

Provide additional context to as part of the prompt

Fine tuning

Provide additional training to LLM

### **Technique** Issues

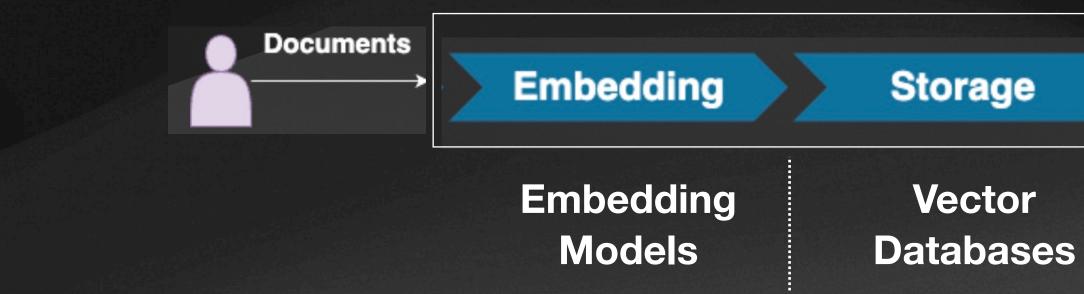
he	<ul> <li>Need to be very precise</li> </ul>
	<ul> <li>LLM may choose to ignore directions</li> </ul>
es	<ul> <li>Adds additional latency to every LLM call</li> </ul>
	<ul> <li>Adds cost and Is not always correct</li> </ul>
LLM	<ul> <li>Need to add RAG store to solution</li> </ul>
	<ul> <li>Highly dependent on quality of data provided</li> </ul>
	<ul> <li>RAG may not provide enough info to get answers</li> </ul>
an	<ul> <li>Very expensive when compared to other methods</li> </ul>
	<ul> <li>Time consuming, could degrade LLM</li> </ul>

### There are no techniques that guarantee chatbot answer is 'good'

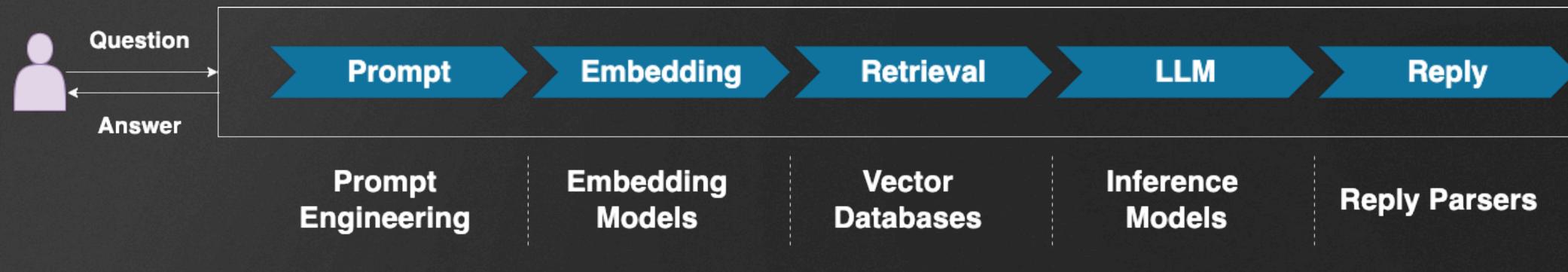


## RAG makes Al be more personal

### Step 1: Data load



#### Step 2: Answer retrieval





### RAG is fallible and thus the arrival of its friends

#### Simple RAG -

#### Context Stuffing

Cache Aug. Gen.-

Goal: Get more precise and useful answers

And there are so many more RAG variants!

#### Graph RAG

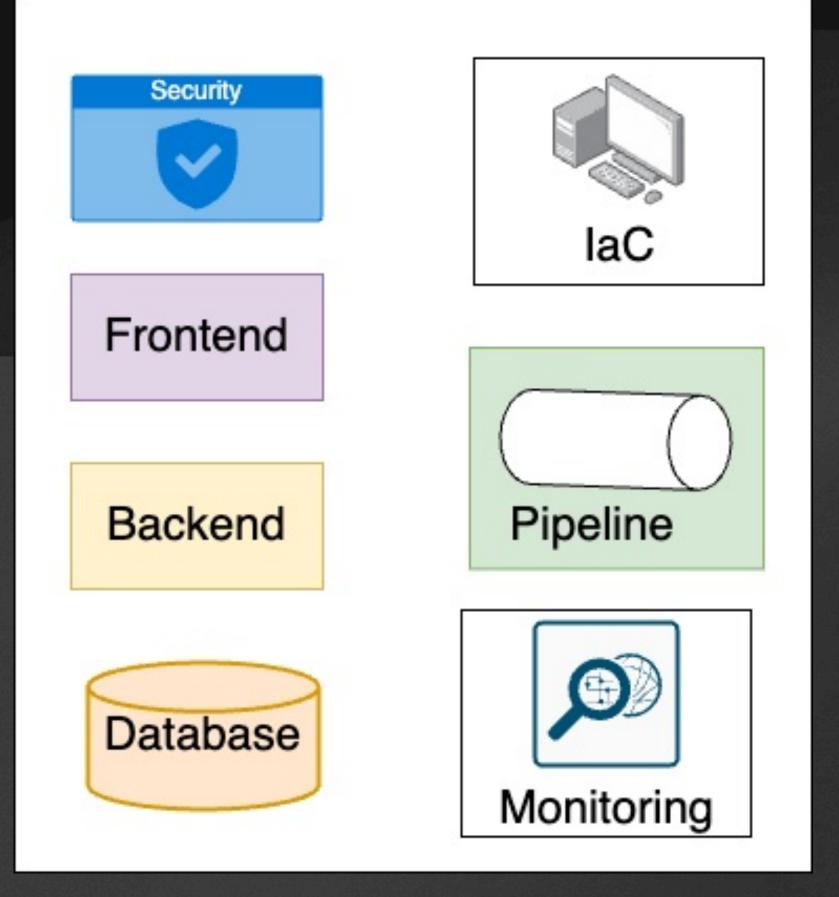
### Multi-modal RAG

### Hybrid RAG

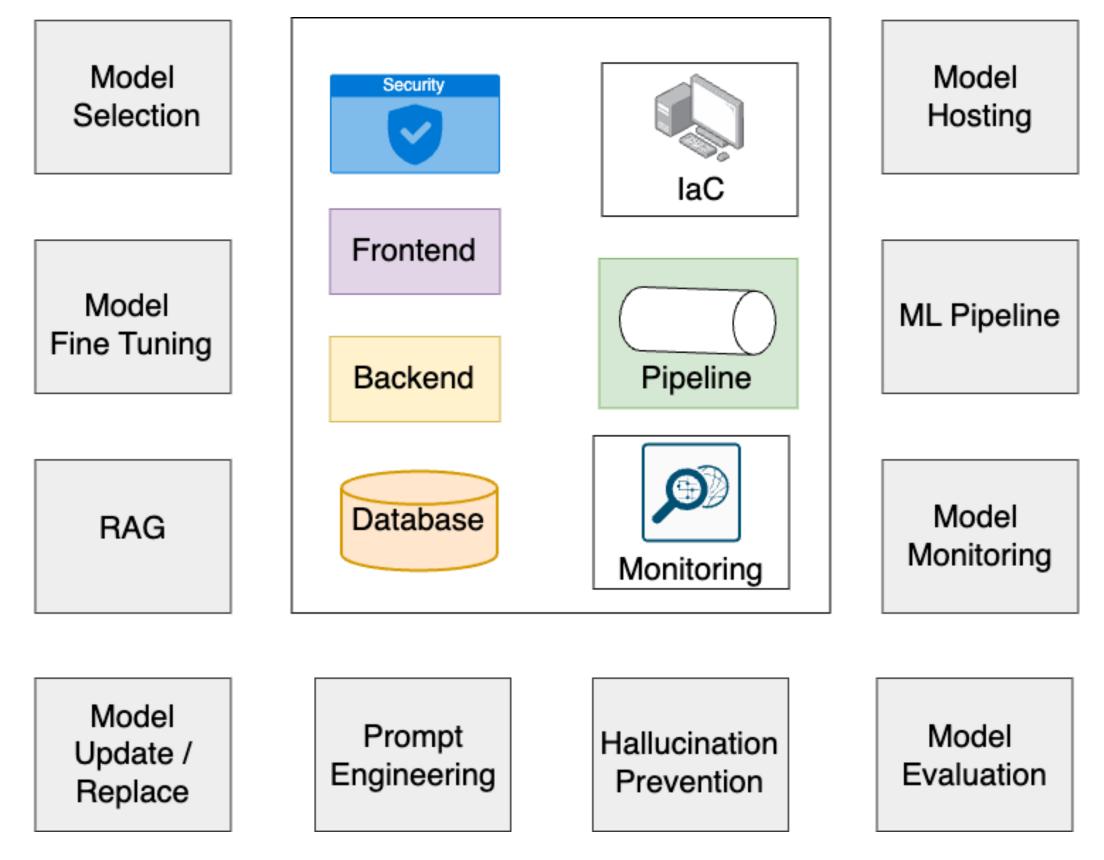


### X AI models increase your app complexity

### Typical app dev components



### Typical AI app dev components





# Choose your models wisely



#### Text Generation

- Hundreds of models available
- Performance varies widely
- Switching LLMs requires re-evaluation of prompts

### Image generation

- Quality of images dependent on prompt & model
- Text to image, image to image and others
- More expensive that LLMs



#### Video generation

- Generation can take minutes
- Text to video, image to video and others
- More expensive than image generation

#### Sound generation

- Not as common or widely used
- Less models to choose from

#### Music generation

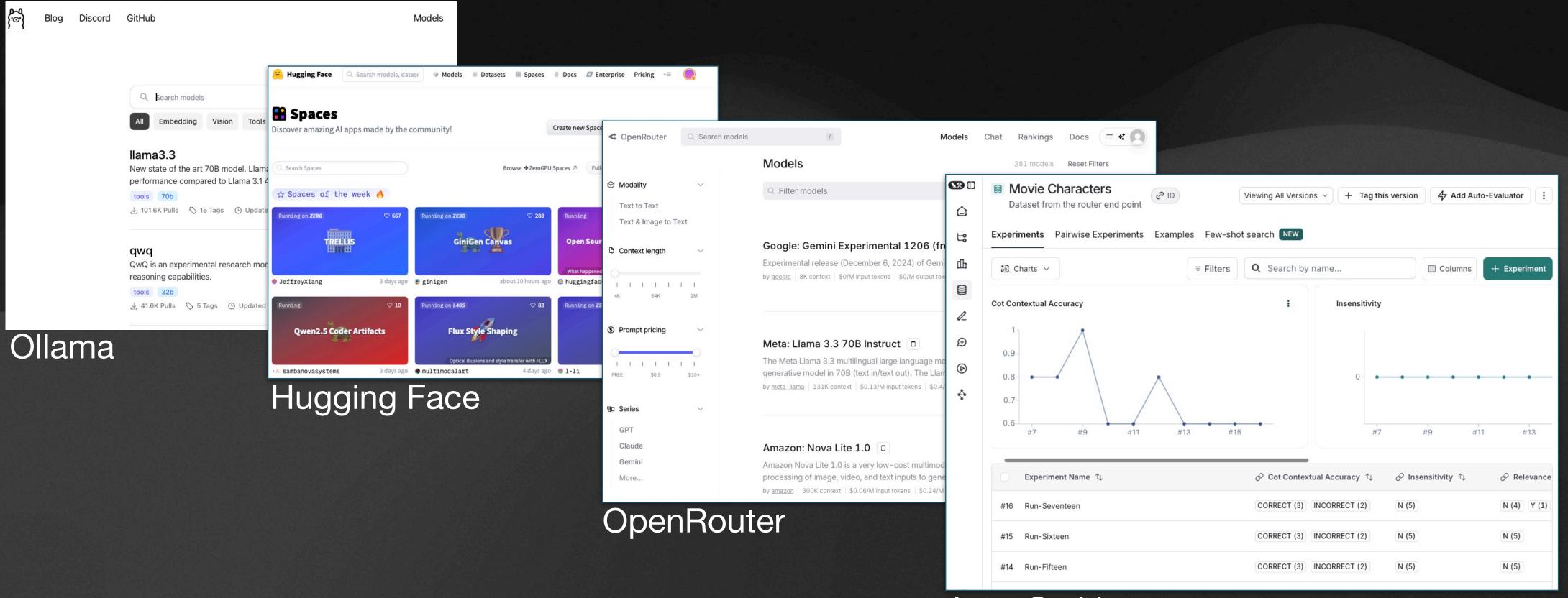
- Quality depends on prompt and/or original audio
- Text to Audio, audio to audio

### Speech generation

- Must cleanup input text
- Performance varies widely. Newer models are great
- Trade off between speed and intonation/quality



### Evaluate models & choose the best for the use case



### Evaluation must take place at each step of the way

#### LangSmith



### Externalize your prompts

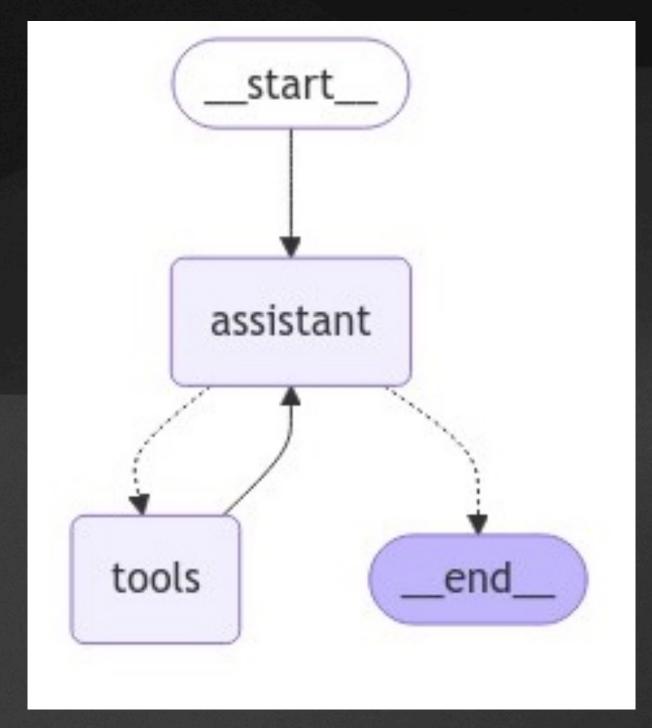
	+	New Prompt 🛛 🗸	8	Use Cases	~
~				Agent simulations	58
	LangChain Hub			Agents	535
L:	Explore and contribute prompts to the community hub			Autonomous agents	s 98
·	Explore and contribute prompts to the community hub			Chatbots	322
đh	Search prompte use cases models	Q		Classification	83
0	Search prompts, use cases, models	4		Code understandi	66
				Code writing	92
R	Top Favorited Top Viewed Top Downloaded Recently Updated		Evaluation	111	
			Extraction	182	
Ð				Interacting with A	96
	ChatPromptTemplate Summarization English openai:gpt-3.5-turbo	$\odot$		Multi-modal	24
$\bigcirc$	hardkothari/prompt-maker			QA over docume	349
~				Self-checking	59
÷	Convert your small and lazy prompt into a detailed and better prompts with this template.			SQL	28
	{×} Prompt • Updated a year ago • ♡ 204 • ③ 69.3k • 🛃 25.1k • - 0-1			Summarization	262
				Tagging	34
	ChatPromptTemplate QA over documents English	$\odot$	$\langle \rangle$	Туре	~
	rlm/rag-prompt			ChatPromptTe	4655
	This is a prompt for retrieval-augmented-generation. It is useful for chat, QA, or other applications	that		StringPromptTe	1378
	rely on passing context to an LLM.			StructuredPrompt	234
	{×} Prompt • Updated 6 months ago • ♡ 175 • ④ 250k • 🛃 19.3M • - 3				
			0	Language	~
	Agents Interacting with ABIs ChatBrowntTemplets metallams 2,70h shot			Chinese	95
	Agents Interacting with APIs ChatPromptTemplate meta:llama-2-70b-chat	$\odot$		English	1122
	homanp/superagent			French	50

### Prompts should not be embedded in the codebase

### Externalizing prompts facilitates:

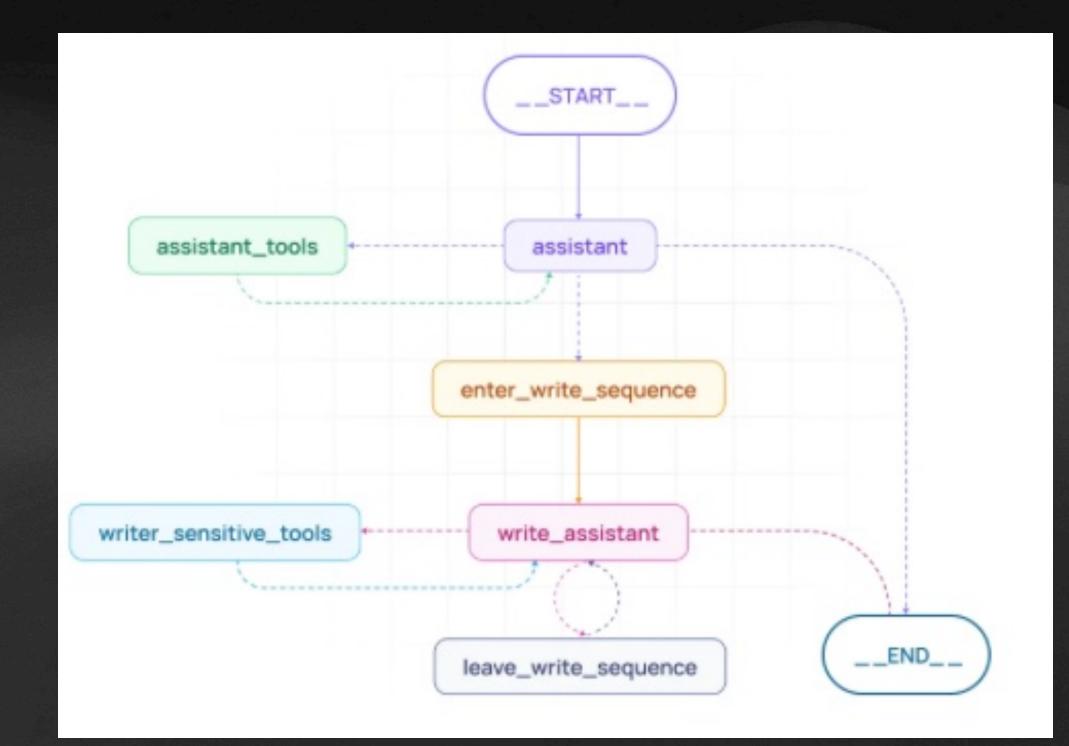
- Expert input
- Future proofing
- Faster development

### Use agents to go beyond the chatbot



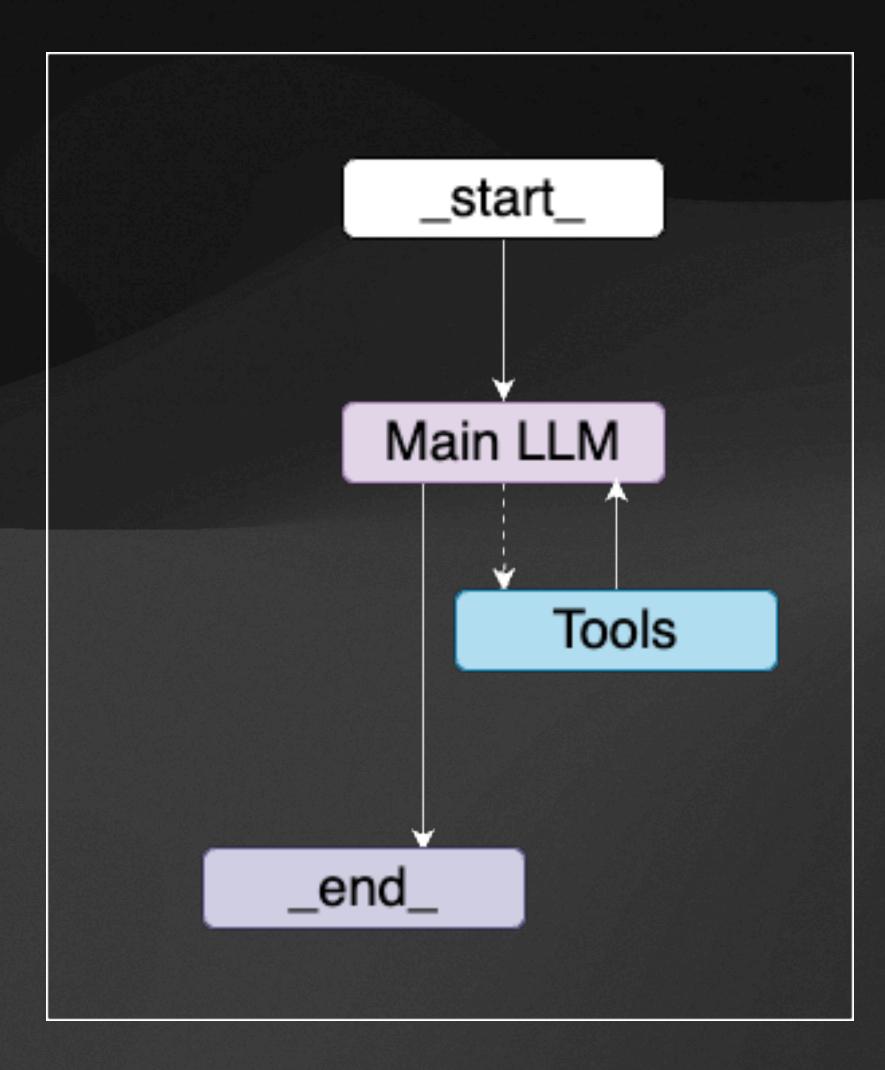
#### Simple tool calling agent

Agents allow LLMs to interact with the real world



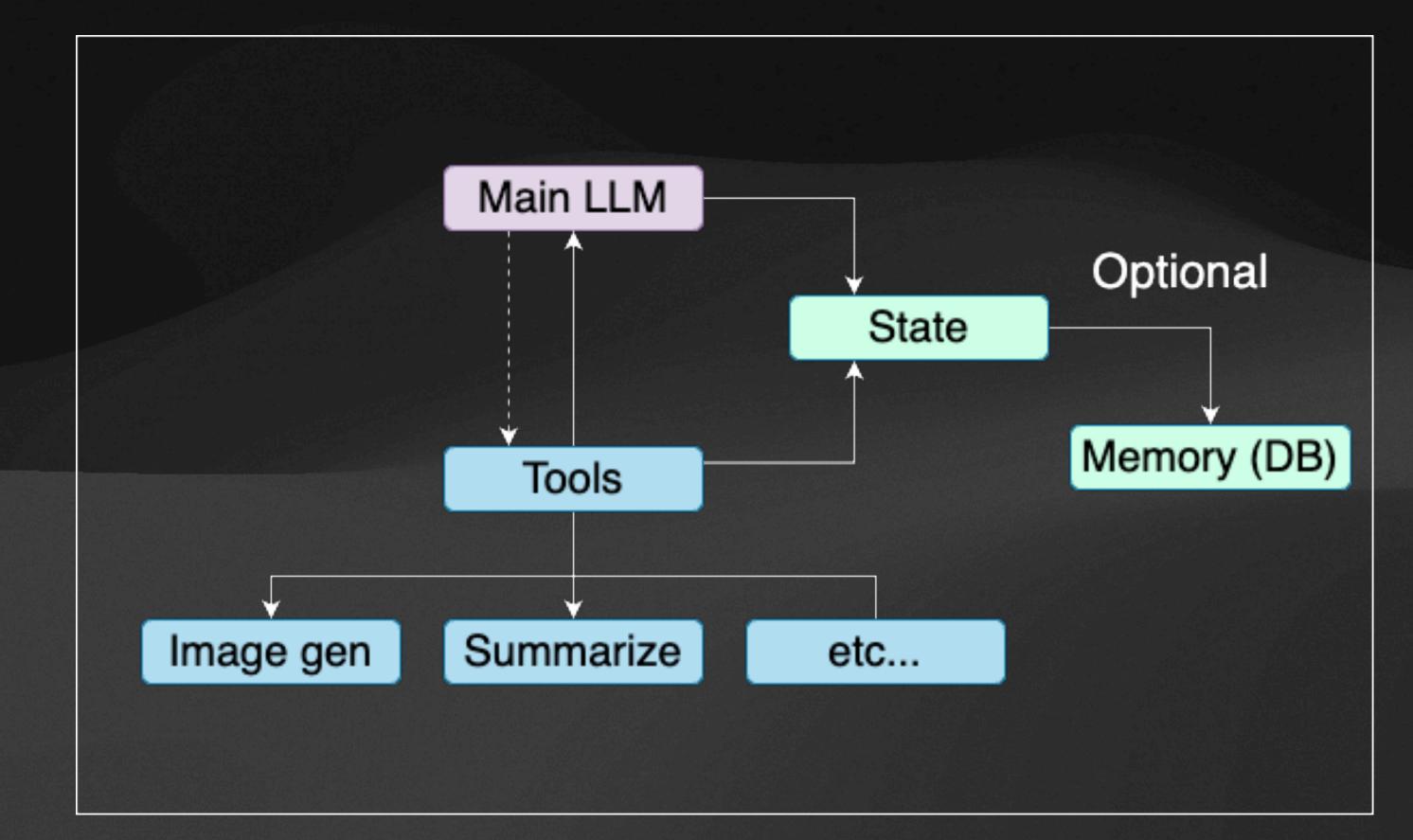
#### Simple assistant agent

### Interacting with the world via a reAct agent



User calls agent Agent calls main LLM LLM use tools to perform task(s) Agent returns result

### Tools: fancy name for functions the agents can use



Tool can be as simple as a function that adds 2 numbers

### Tools are called by LLM

Tools share info via a state



## Memory can store helpful information

- User Settings
- Personalization
- Preferences
- **Conversation facts**



Use memory to help smooth out user interactions

Beware of storing user's personal info

 May subject you to additional regulations

- Could make you a target of a malitious actor

Store what you need and nothing else

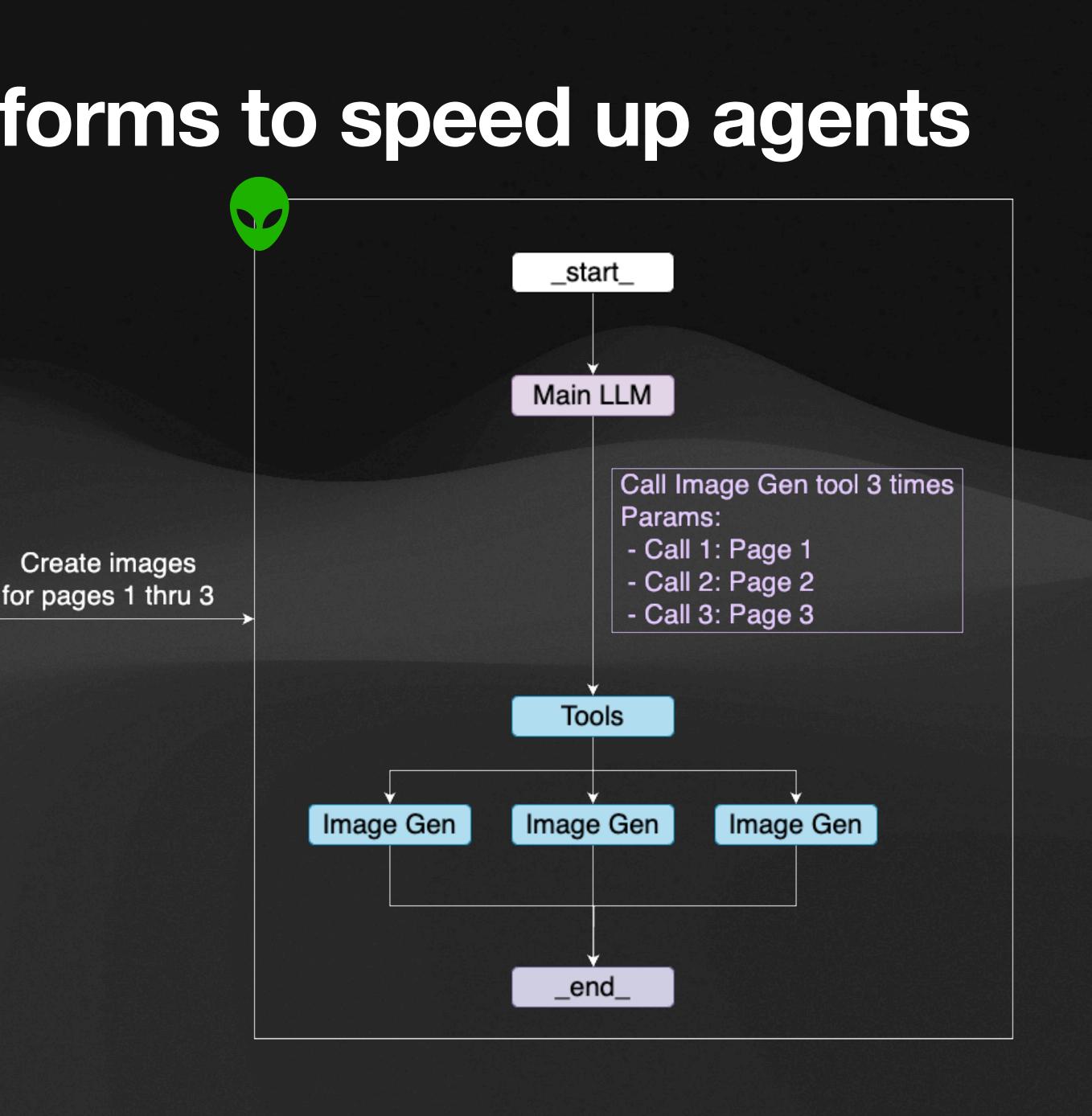


### Take advantage of platforms to speed up agents

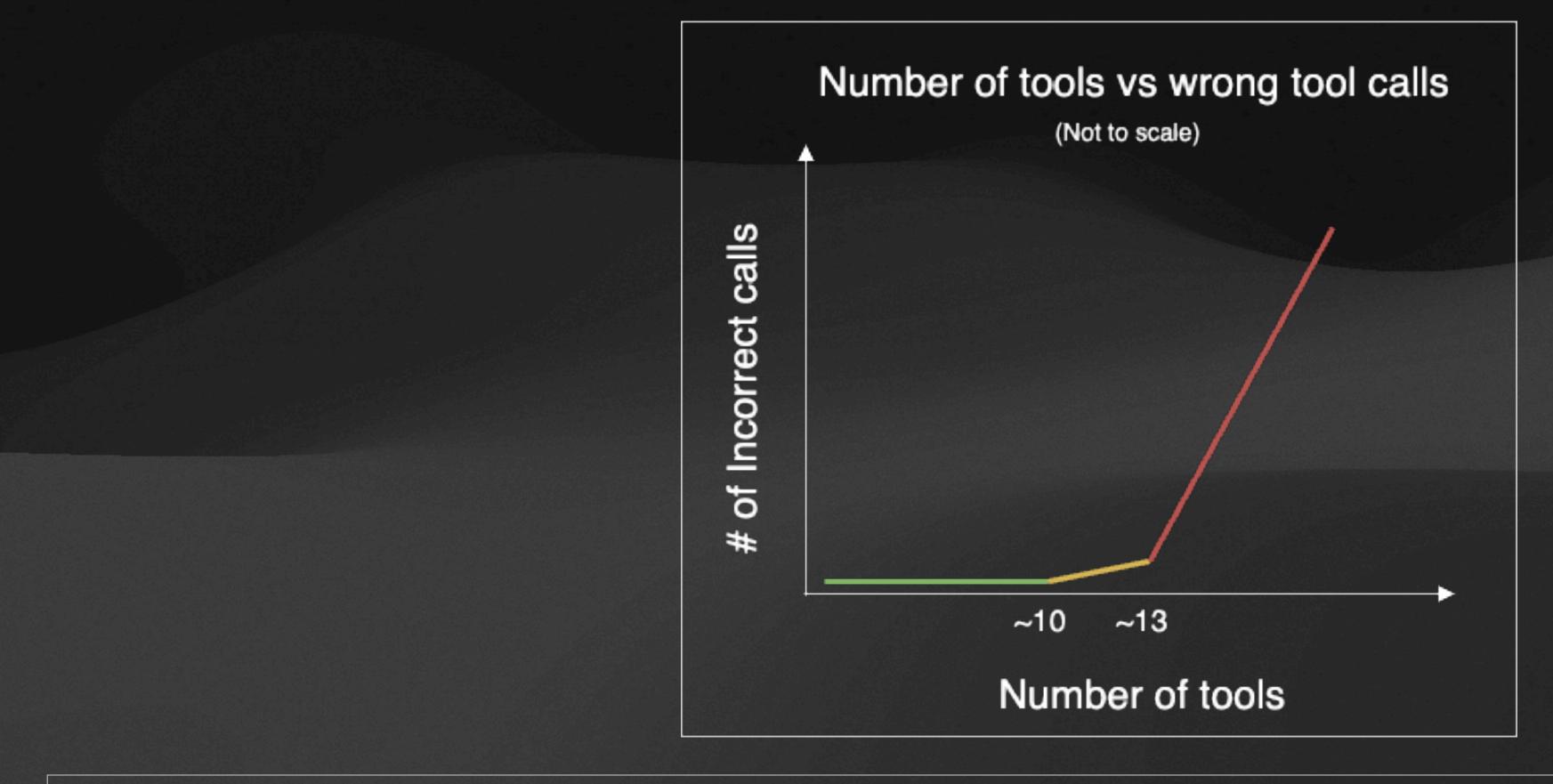
### Models are the biggest source of latency

### Platforms like LangGraph allow complex chaining of models & code

- Concurrent calls
- Branching



## Crucial: # of functions and their descriptions



LLMs decide which tools to call based on their names, parameters & descriptions They are not always right !



### Be precise on your tools metadata for better accuracy

#### Bad

#### def myFunction(a, b)

#### def generate(a: str, b: str)

Precise metadata improves chances of LLM doing the right thing

.....

#### Better

def generate\_image\_for\_location
 (location\_name: str, image\_quality:
 ImageQuality)

Generate an image for a location

Args:
 location\_name: name of the location
 image\_quality: quality of the image to be
generated
"""

### LLMs can follow directions on the metadata



class ImageQuality(Enum):
 """Quality of the image to be generated. Use low as default"""
 LOW = "low"
 MEDIUM = "medium"
 HIGH = "high"
 EXTREME = "extreme"

Note the default quality to be used by LLM



### When something goes wrong, LLMs go off the rails

# If a tool fails, LLM will try to call all other tools to fix the issue using brute force

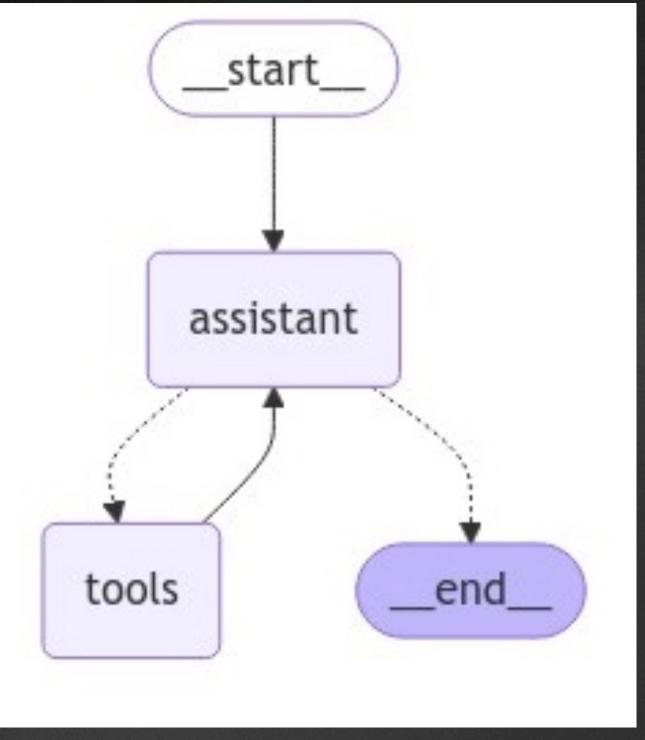
# Always check how many steps it takes to solve users' request

# Set the maximum number of iterations per call for the agent



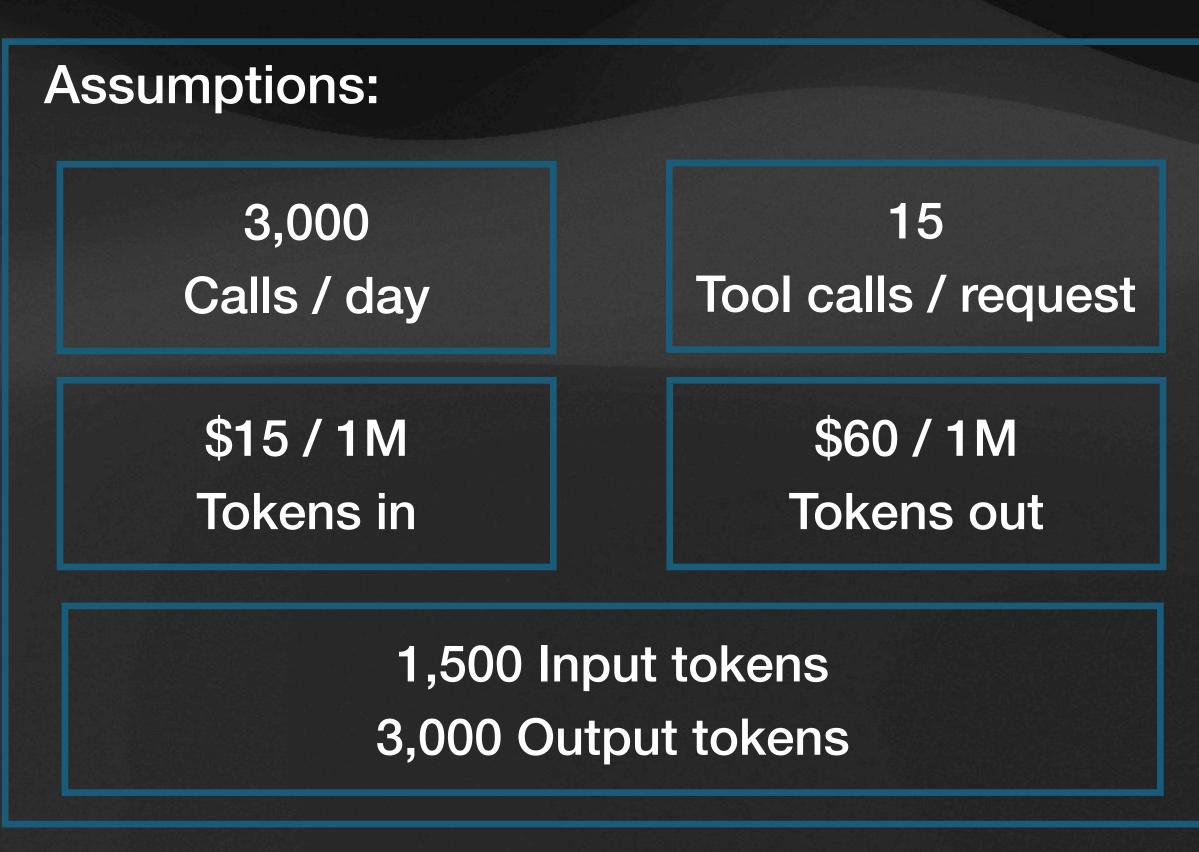
## Be aware of the cost when running agents

Call Center example: Agent that uses OpenAl o1 as the LLM



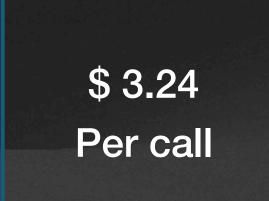
#### Simple tool calling agent

Pricing source: OpenAI API pricing webpage as of Mar 31st, 2025





### Costs can add up quickly





- 0.000015 \*1500 Input token cost
- 0.00006 \* 3000 Output token cost
- (16 \* (0.000015 \* 1500 + 0.00006 \* 3000)) =\$ 3.24

3.24 \* 3000 calls/day = \$9,720



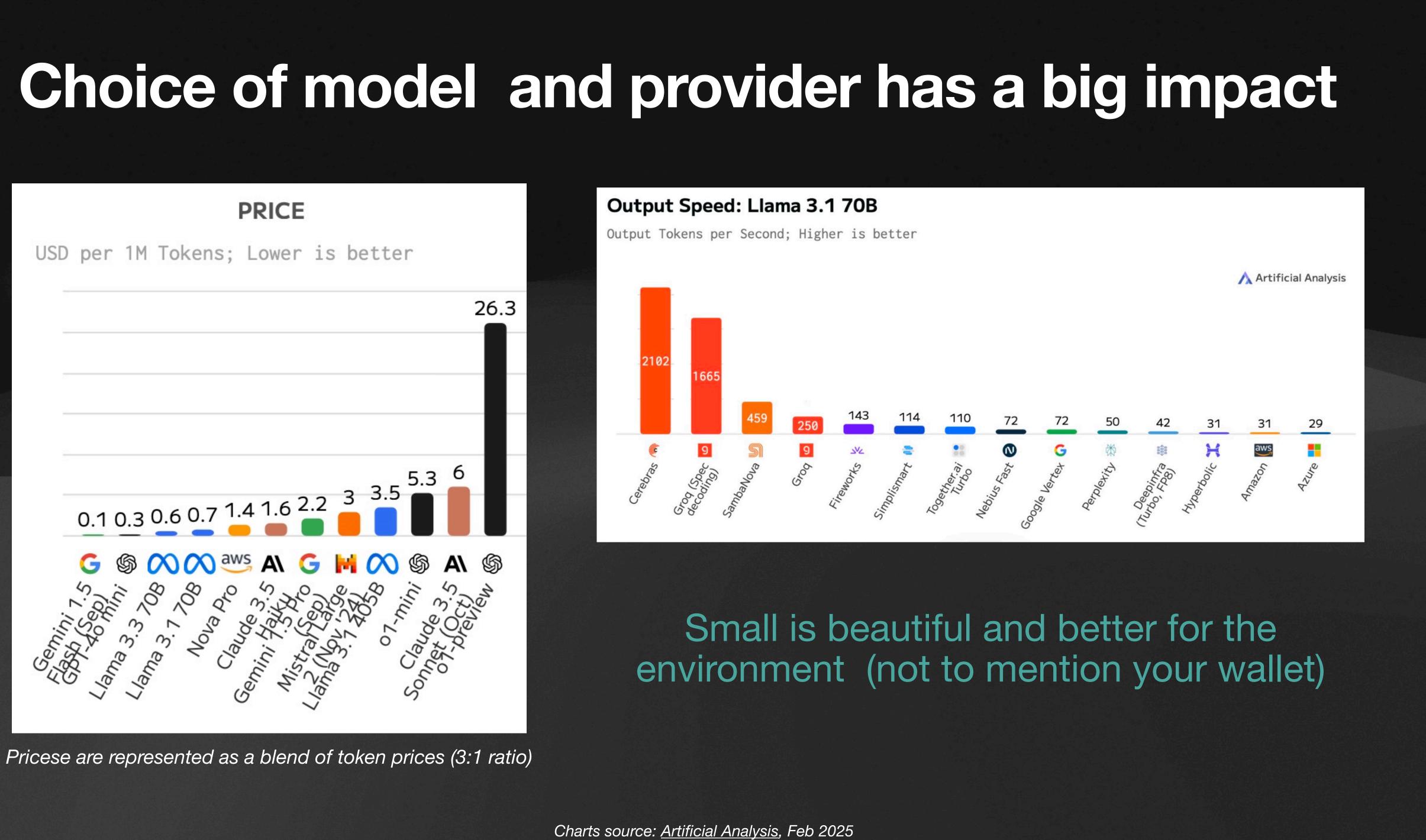
#### Example call center agent using OpenAl o1 as the LLM

\$ 9,270 Per day

### \$291,600 per month

9,270 \$/day \* 30 days = **\$ 9,720** 





## The right tools for the job can minimize cost

Example call center agent using Llama 3.3 70B as the LLM running on Groq

\$ 0.052 Per call

- 16 o1 calls (1per tool call + initial call)
- 0.00000059 \*1500 Input token cost
- 0.00000079 \* 3000 Output token cost
- (16 \*(0.00000059 \*1500 + 0.00000079 \* 3000)) = \$ 0.052

0.052 \* 3000 calls/day = \$156

Pricing source: Groq API pricing webpage as of Mar 31st, 2025



156 \$/day \* 30 days = **\$ 4,680** 



## The right tools for the job can minimize cost

Llama 3.1 8B, Llama 3.3 70B, OpenAl o1



Pricing source: Groq API and Open AI pricing webpages as of Mar 1st, 2025 Open AI o1: \$15 in & 60 out per 1M , Llama 3.3 70B: \$0.59 in & 0.79 out per 1M , Llama 3.1 8B: \$0.05 in & 0.08 out per 1M

\$4,680 Per month





### Remember to keep your LLM protected at all times

Having a chatbot as your input source increases the potential attack vectors to your data



Treat LLMs inputs and outputs just as potential bad actors trying to access your data





### Increase Agent success rate with a classifier

#### Reduce the number of tools needed by your LLM at any one point

Break up the task into smaller problems by putting a classifier in front of your agent. Have the classifier direct requests to different agent nodes instead of overloading your agent with too many tools



### Stop tool calling rampage by escaping the loop

Build an escape clause into the LLM tool calling loop If your app throws an error, ensure the LLM is not called again. Just report error to the user directly

Stops sticker shock and makes the app more user friendly



### Make sure you have fallbacks

LLM endpoints are pretty robust, but not infallible

What do you do when a provider goes down?

### Each Agent call can result in a dozen of LLM calls

#### Each LLM call comes with an additional cost

When using LLM endpoints, each user added to app increases the cost linearly. Economies of scale do not apply.

When renting GPUs, the cost of each additional GPU is pretty high and GPUs are hard to come by

Hard to come up with an appropriate pricing structure



### Sometimes it is better to run agentic workflows

No need for an orchestrating LLM to decide which tool to call when the task path is pre-determined

Can still use LLMs to perform the work

Be nice to your wallet



### Use agents and LLMs only when you need them

AI APIs providers are convenient and efficient but cost is linear per user



Always keep an eye on your wallet and budget

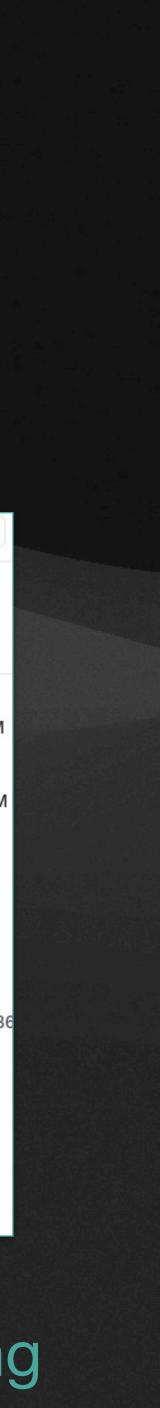
GPUs are really expensive and hard to optimize for multi tenancy



# Get proper observability for your Agents

▲ Personal > Tracir	ng projects 🔸 比 Test P	» <sup>KX</sup> V A	Comp	are				
😫 Test Proje	ct	TRACE	⊘ /agent Image: Add to v 1	R				
च 1 filter  ☐ Las	Monitor Setup	Filter       H       V         (agent)       (agent)       (agent)         (agent)       (agent)       (agent)       (agent)         (agent)       (agent)       (agent)       (agent)         (agent)       (agent)       (agent)       (agent)         (agent)       (agent)       (agent)       (agent)         (agent)       (agent)       (agent)       (agent)         (agent)       (agent)       (agent)       (agent)       (agent)         (agent)       (agent)       (agent)       (agent)       (agent)       (agent)         (agent) <th(agent)< th="">       (agent)       (agent)<th>Run Feedback Metadata 🗇 Run ID 🗇 Trac</th><th>😫 Test I</th><th>→ Tracing projects → <b>ヒ</b>8 Test P <b>Project</b> eads Monitor Setup</th><th>TRACE</th><th>Agent () Playgrou</th><th>Compare and Add to &gt; 1 2</th></th(agent)<>	Run Feedback Metadata 🗇 Run ID 🗇 Trac	😫 Test I	→ Tracing projects → <b>ヒ</b> 8 Test P <b>Project</b> eads Monitor Setup	TRACE	Agent () Playgrou	Compare and Add to > 1 2
0	<ul> <li>Name</li> <li>/agent</li> <li>/agent</li> </ul>	<ul> <li>Ilama-guard-3-8b 0.34s</li> <li>check_safety 0.00s</li> <li>question_cl 0.80s ∨</li> <li>agent_quest 0.80s ∨</li> </ul>	<u>langserve_endpoint</u> : <u>"invoke"</u> <u>langserve_version</u> : <u>"0.3.0"</u> <u>useragent</u> : <u>"node"</u> <u>current_user_id</u> : <u>"user_2hkJy9vhu5Wo6rY1eVBwLdRHOrY"</u>	च 1 filter     →	<ul> <li>Last 7 days</li> <li>Root Runs</li> <li>Name</li> <li>/agent</li> </ul>	<ul> <li>/agent 1.58s ()</li> <li>IIm_guard 0.92s ()</li> <li>IIama_guard 0.92s ()</li> <li>IIama-guard 0.92s ()</li> <li>IIama-guard-3-8b ().43s</li> <li>check_safety 0.00s</li> </ul>	Error Copy          2 validation errors for ChatGroq          model_name         Input should be a valid string	START TIME 12/06/2024, 12:20:01 PM END TIME 12/06/2024, 12:20:02 PM
□	<ul> <li>/agent</li> <li>/agent</li> <li>/agent</li> </ul>	<ul> <li>Pro Prompt 0.00s</li> <li>Ilama-3.1-8b-inst 0.39s</li> <li>check_classifica 0.00s</li> <li>chatbot 0.73s </li> </ul>	revision_id: "434537f-dirty"         RUNTIME ^         Iangchain_core_version: "0.3.9"         Iangchain_version: "0.3.2"         library: "langchain-core"	B Prompt 0.00s   3.1-8b-inst 0.39s  classifica 0.00s   langchain_core_version: "0.3.9"   langchain_version: "0.3.2"   bt 0.73s     library: "langchain_core"     library: "langchain_core"	<ul> <li>⊘ question_cl 0.62s ∨</li> <li>⊘ agent_questi 0.61s ∨</li> <li>Pro</li></ul>	<ul> <li>input_value=0.0, input_type=float]</li> <li>For further information visit</li> <li>https://errors.pydantic.dev/2.9/v/ string_type</li> </ul>	TIME TO FIRST TOKEN N/A STATUS () Error	
	<ul> <li>/agent</li> <li>/agent</li> </ul>	<ul> <li>Ilama3-70b-8192 0.67s</li> <li>tools_condition 0.00s</li> <li>tools 0.60s </li> </ul>	library_version: "0.3.9" platform: "macOS-15.1.1-arm64-arm-64bit" py_implementation: "CPython"	□ > □ > □ >	<ul> <li>/agent</li> <li>/agent</li> <li>/agent</li> </ul>	<ul> <li>check_classifica 0.00s</li> <li>chatbot 0.01s</li> </ul>	temperature Input should be a valid number, unable to parse string as a number [type=float_parsing, input_value='llama3-70b-8192', input_type=str]	TOTAL TOKENS 777 tokens / \$0.0001023 LATENCY © 1.58s
Agent run traces in LangSmith				□ <b>&gt;</b>	<ul><li>/agent</li><li>/agent</li></ul>		For further information visit https://errors.pydantic.dev/2.9/v/ float_parsingTraceback (most recent call last):	TYPE Chain

#### Tracing model output, specially with agents, is hard without the necessary tooling





### **Questions**?

#### Juan Peredo

linkedin.com/in/juanperedotech

