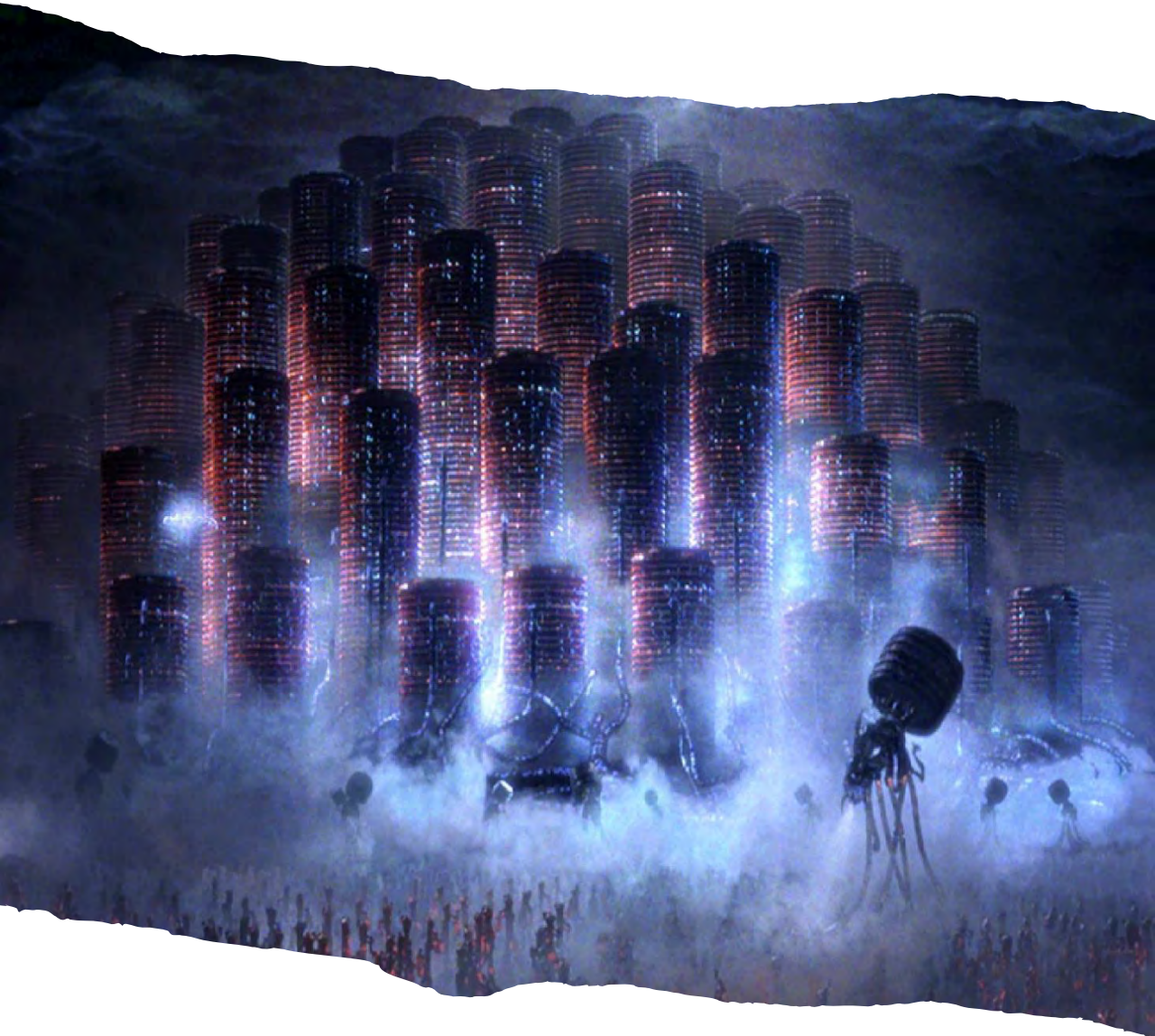
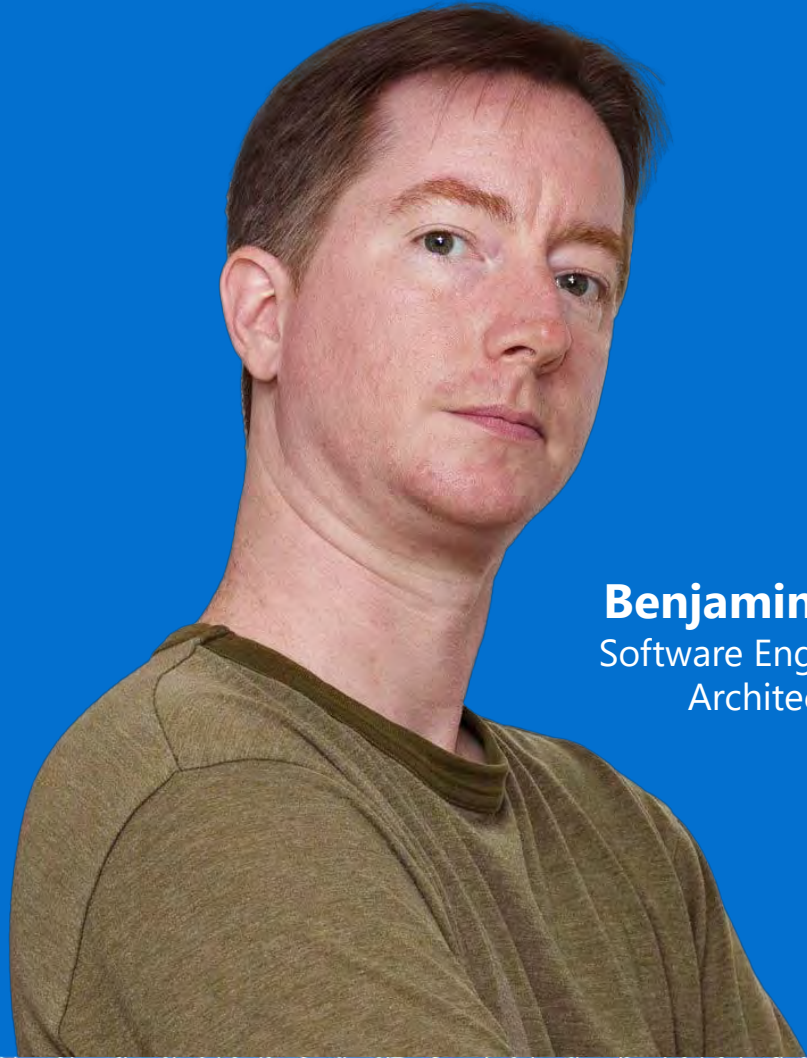


Artificial Intelligence - Hope vs. Hype





Artificial Intelligence - Hope vs. HYPE



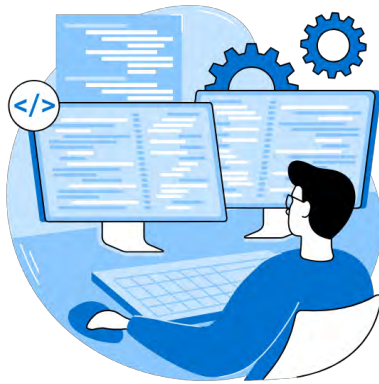
Benjamin Hall
Software Engineer /
Architect

MEET THE SPEAKER

- Purdue University Alumni CGT
- Freelance Developer
- Software Engineer / Architect
- Full Stack Developer and Software Architect – Theoris Solutions
- Technologist
- Winner of 2008 Innovation Award TechPoint
- Broad AI Experience

ABOUT THEORIS

SOLUTIONS



Custom App Dev
& Maintenance

Code & Architecture
Review

Data Management
& Visualization

SERVICES



IT Staffing

Engineering Staffing

Quality & Validation
Staffing

STRATEGY



IT Project Management

QA & Technical Writing

Business Process Review

AGENDA

- The Current AI Landscape
- Generative AI In Depth
- Agentic AI Explained
- AI in Software Development
- Vibe Coding: What's Real vs. Risky
- Q&A



THE ARTIFICIAL INTELLIGENCE AGE

What People Think AI is:

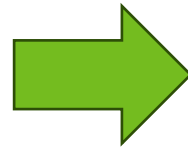


What AI Really is:



THE ARTIFICIAL INTELLIGENCE AGE

AI is NOT going to replace people. It's going to make life easier.



** What the Mechanical Age did for the physical laborer, the AI Age will do for the Mental Laborer.*

EMERGING TECHNOLOGY LIFECYCLE

DISCOVERY



We Discovered Something!

- Media Attention
- Small Level of Adoption
- Buzz begins...

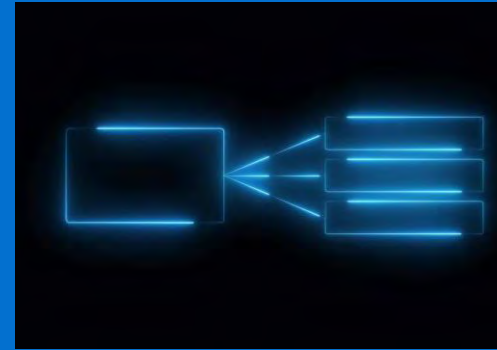
BLUR



A RACE to Understand and Control

- Massive Attention
- Large Scale Adoption
- Some Panic!

DIVERGENCE



Use Cases Diverge

- Technology standardizes
- Adoption becomes realistic
- Panic falls away...

FOCUS



Use Cases Solidify

- Technology is routine
- Adoption is ubiquitous
- Buzz becomes "boring"

YOU ARE HERE

AI TYPES & THEIR ROLE IN IT

1. **Rule-based AI:** Follows predefined rules
2. **Predictive AI:** Forecasts outcomes from data
3. **Generative AI:** Creates code and content
4. **Agentic AI:** Acts autonomously to complete tasks

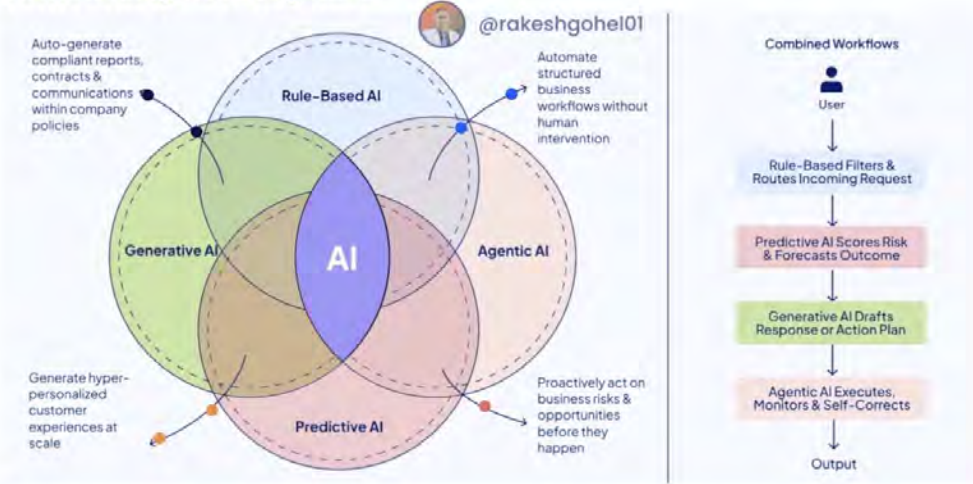
We're focusing on how Generative AI applies to software development and IT teams.

Understanding the Types of AI

Understanding the types of AI and how to combine them for best workflow

🔍 Rule based AI	📊 Predictive AI	🌱 Generative AI	🤖 Agentic AI
What it means? AI that follows predefined rules, logic set by your organization to ensure consistent, compliant outcomes	What it means? AI that analyzes your enterprise data to forecast risks, demand and opportunities before they impact the business	What it means? AI that creates business-ready content, code and strategies tailored to your enterprise context and brand voice	What it means? AI that acts as an autonomous digital employee – planning, deciding and executing complex multi-step tasks
What it can do? <ul style="list-style-type: none"> Automate approval workflows Enforce regulatory & compliance policies automatically Flag policy violations in real-time 	What it can do? <ul style="list-style-type: none"> Forecast revenue, demand Predict employee attrition & hiring gaps Detect fraud & financial anomalies early 	What it can do? <ul style="list-style-type: none"> Draft proposals, contracts Generate code, APIs & technical documentation Personalize client communications 	What it can do? <ul style="list-style-type: none"> Run end-to-end business workflows without human input Research, analyze & deliver insights Coordinate across tools, teams & systems

How do they work together

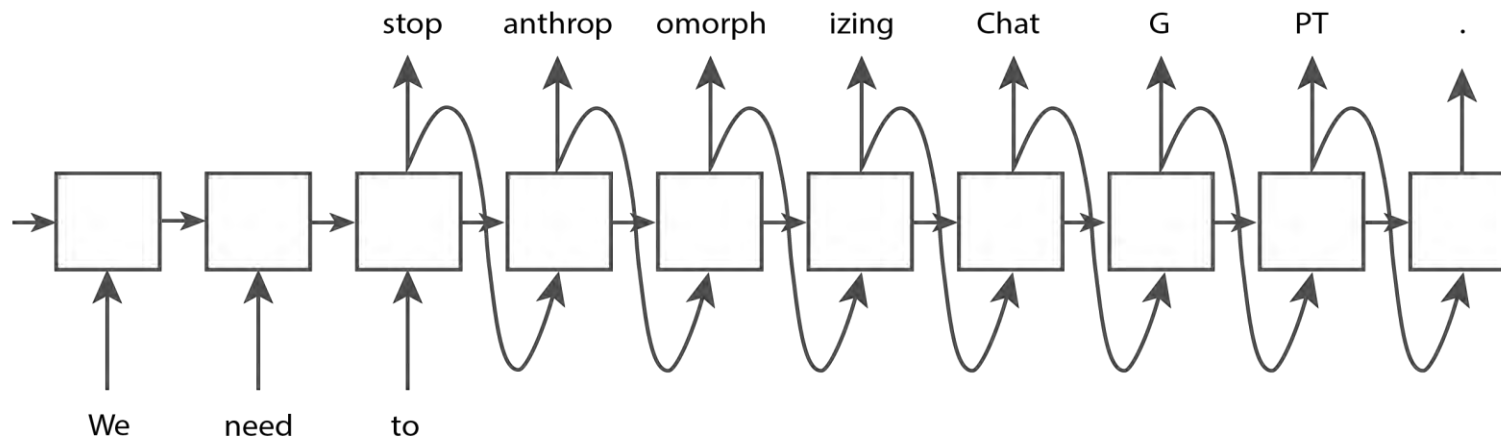
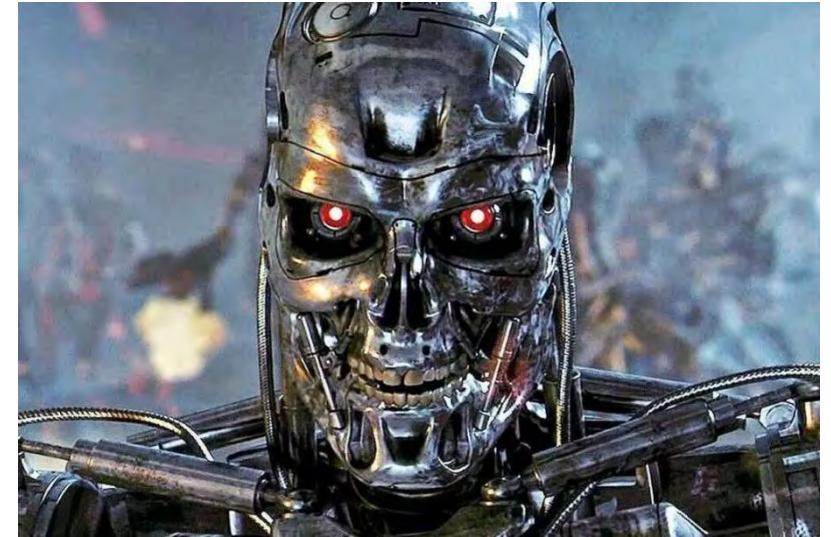


Follow to learn everything about AI Agents | Join my newsletter at newsletter.rakeshgoel.com — @rakeshgoel01



WHAT IS GENERATIVE AI

- It is not **SkyNet**...
- Generative AI uses probability matrices to predict the outcome of a given input.
- It uses complex algorithms to generate text.
- We use words like "Machine Learning", "Training Data" and "Chat" to anthropomorphize technology.
- It's not "**Intelligence**" it's "**Math**".



HOW DOES GENERATIVE AI WORK?

Let's Consider A Generalized Pre-Trained Model (GPT)

Phase 1
(Training)



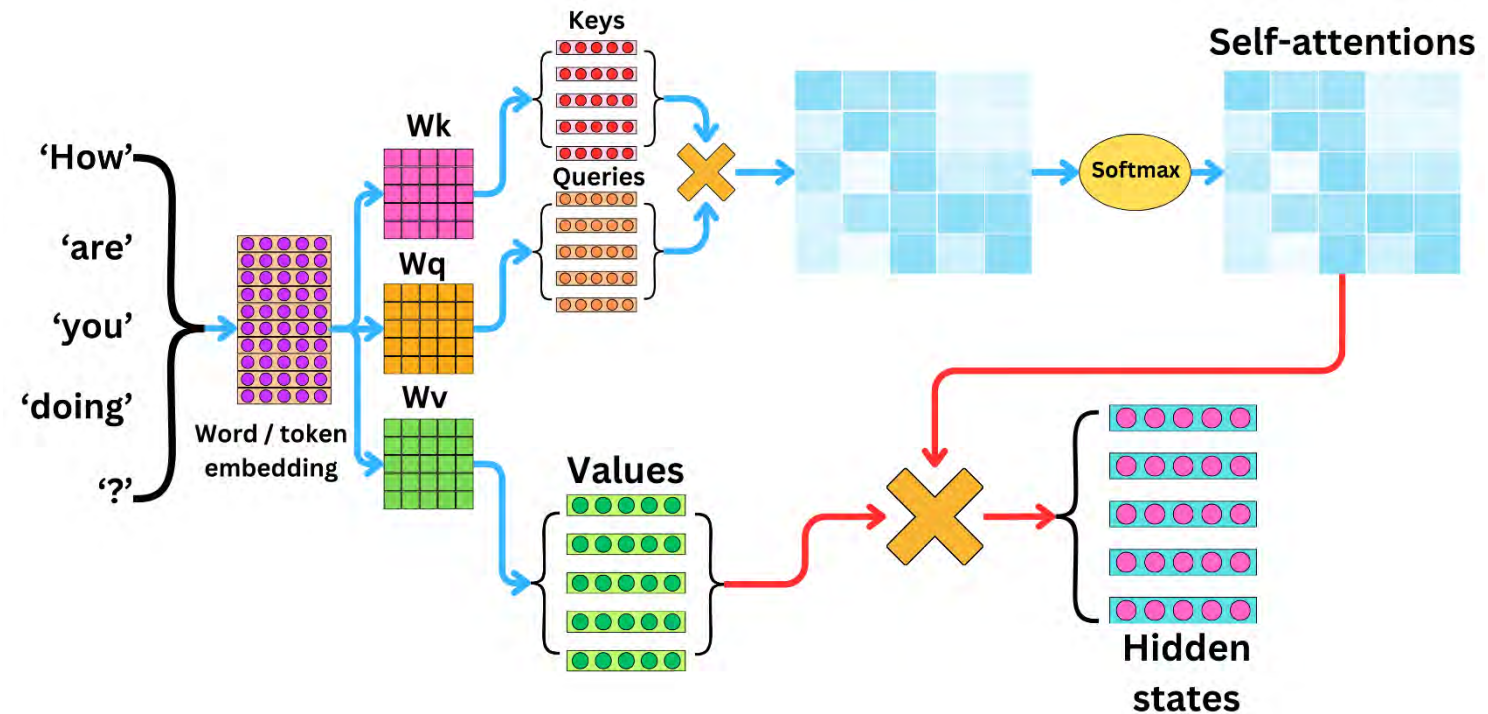
Phase 2
(Implementation)



HOW DOES GENERATIVE AI WORK?

Phase 1: Training a Large Language Model (LLM)

- Training Data is fed into the program and algorithms are applied iteratively.
- Weights are assigned to words – (embeddings)
- These weights also consider words leading and following to further add “meaning” and intent.
- This requires large amounts of DATA and TIME.
- The training process is expensive!*

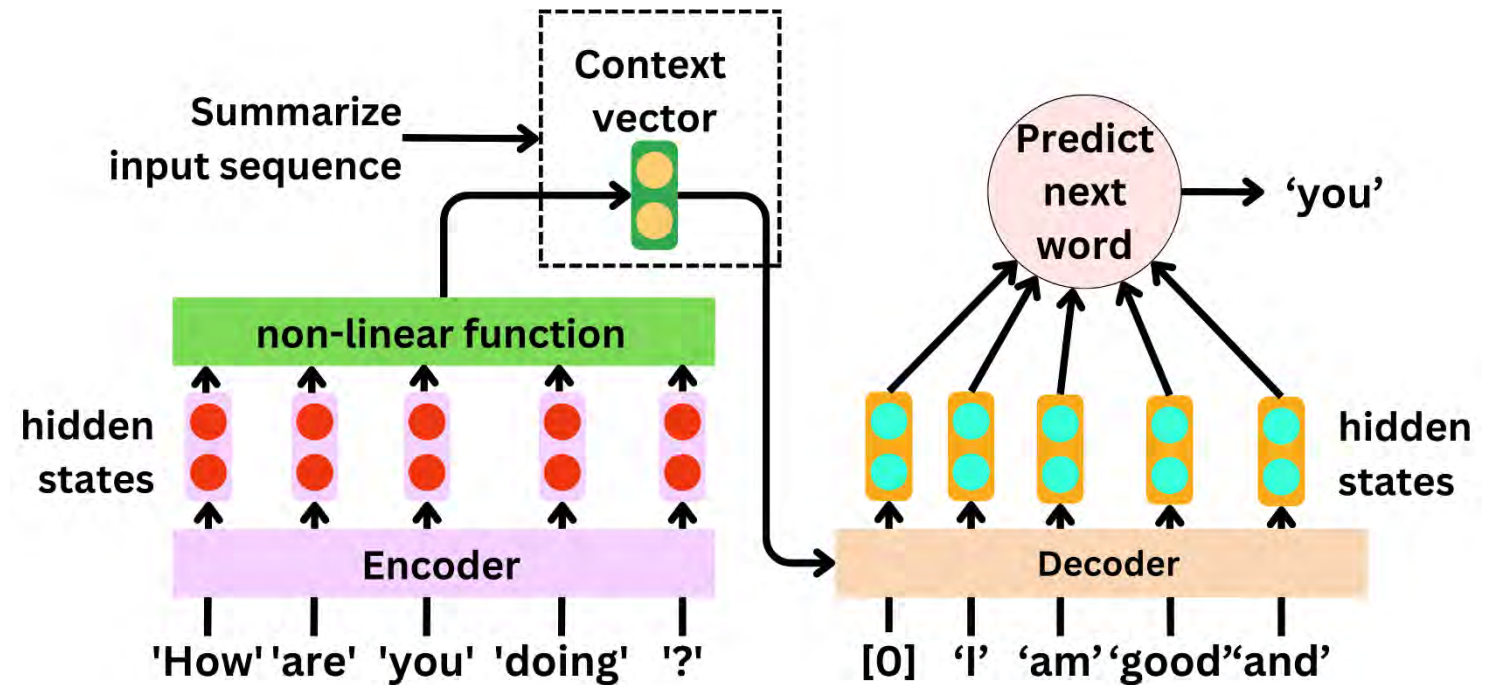


* Which is why LLM files are many GB in size!

HOW DOES GENERATIVE AI WORK?

Phase 2: Using a Large Language Model (LLM)

- User input runs through an encoder and creates embeddings – (like in training)
- The “context” is then sent to the decoder.
- The Process takes the “context” and transforms it into an output through predictive mathematics.
- The more relevant information it has going in, the better it can predict an output that you are asking for.
- Context is KING!



WHAT GENERATIVE AI DOES WELL

- Writing code
- Debugging code
- Writing unit tests
- IT troubleshooting
- Rewording emails
- Generating marketing content
- Help with brainstorming
- Helping to break down complex information
- Analyzing data
- Help with creative writing
- Customer service chats



* CAVEAT: IF the model is trained on the desired content

WHAT GENERATIVE AI DOES NOT DO WELL

- Writing entire applications
- Inferring intent
- Automatically making correct decisions
- Real-World Understanding - (grounding)
- Aesthetics / taste
- Consistency over time - (context window)
- Truth and Reliability - (hallucinations)
- Math
- Security and Trust Boundaries
- Anything outside of its context...

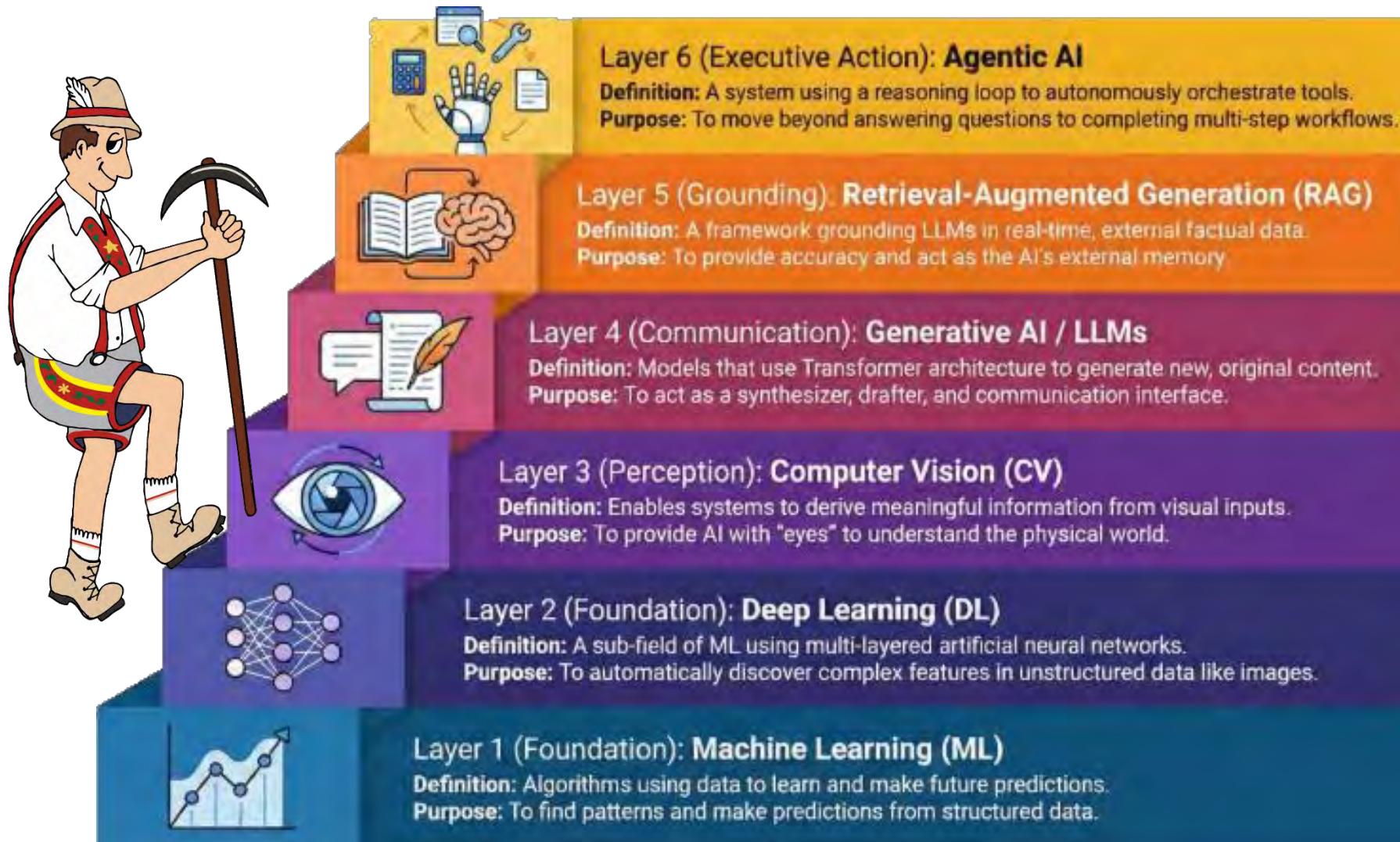


PITFALLS OF GENERATIVE AI

- Over reliance on AI – (\$\$\$)
- Belief that AI is intelligent
- Adopting AI too quickly
- Adopting AI too slowly
- Trusting AI output
- Replacing humans with AI
- Misuse of AI – (crime, fraud, etc.)
- Data security (company data, etc.)



THE MODERN AI STACK

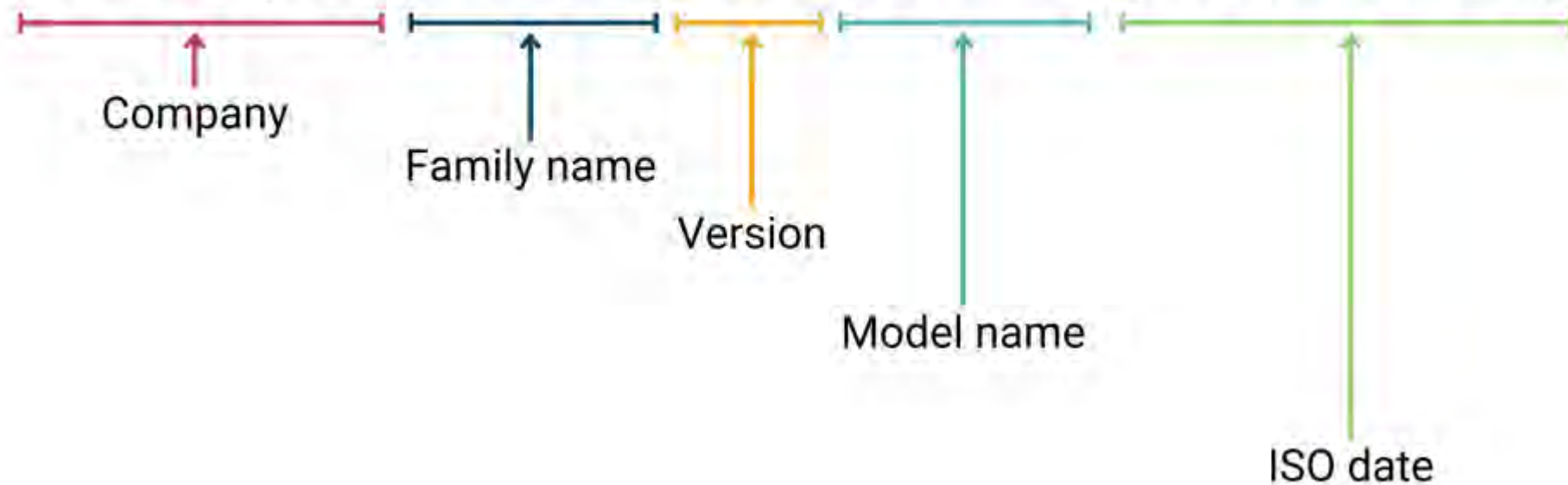


LLM – PROVIDERS



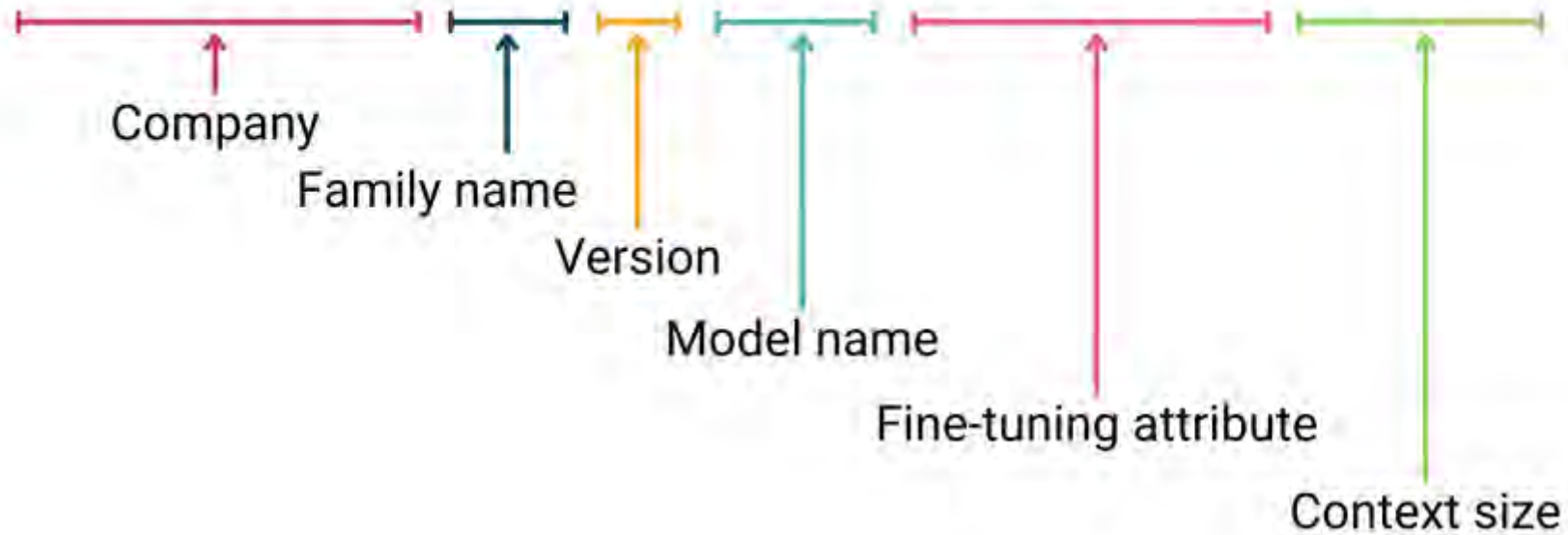
LLM – NAMING CONVENTIONS

Anthropic: Claude v3.5 Sonnet (2024-06-20)



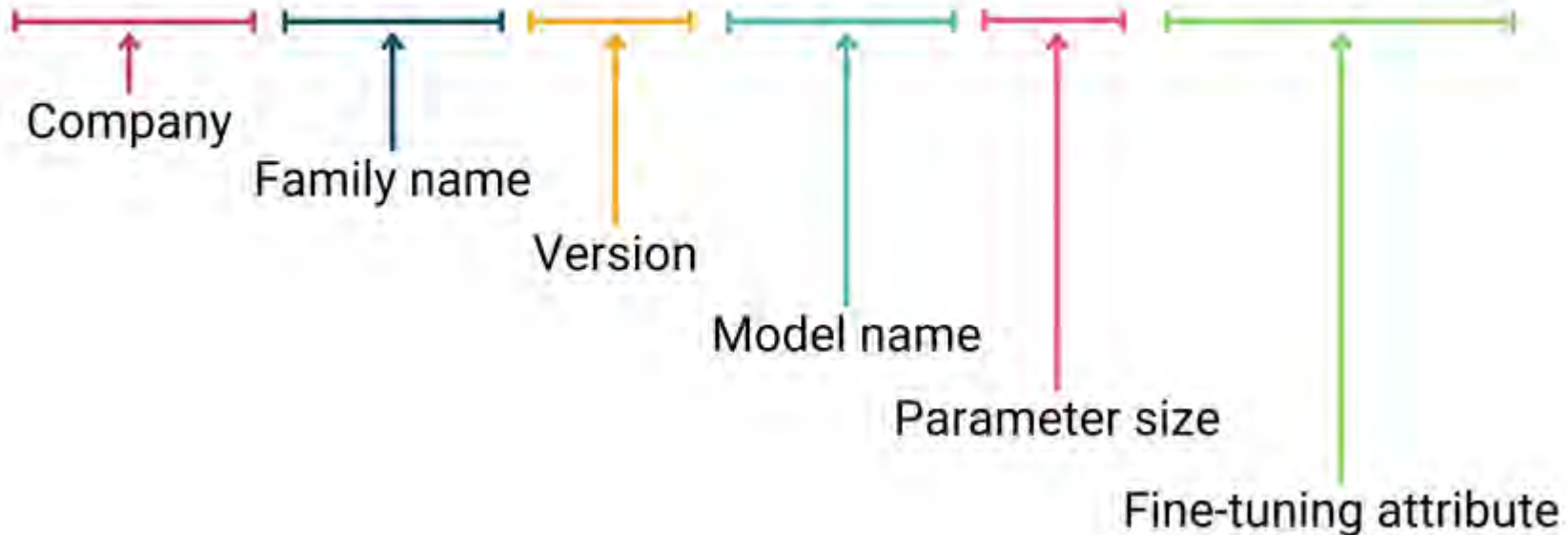
LLM – NAMING CONVENTIONS

Microsoft: Phi v3 Mini (instruct) (128k)

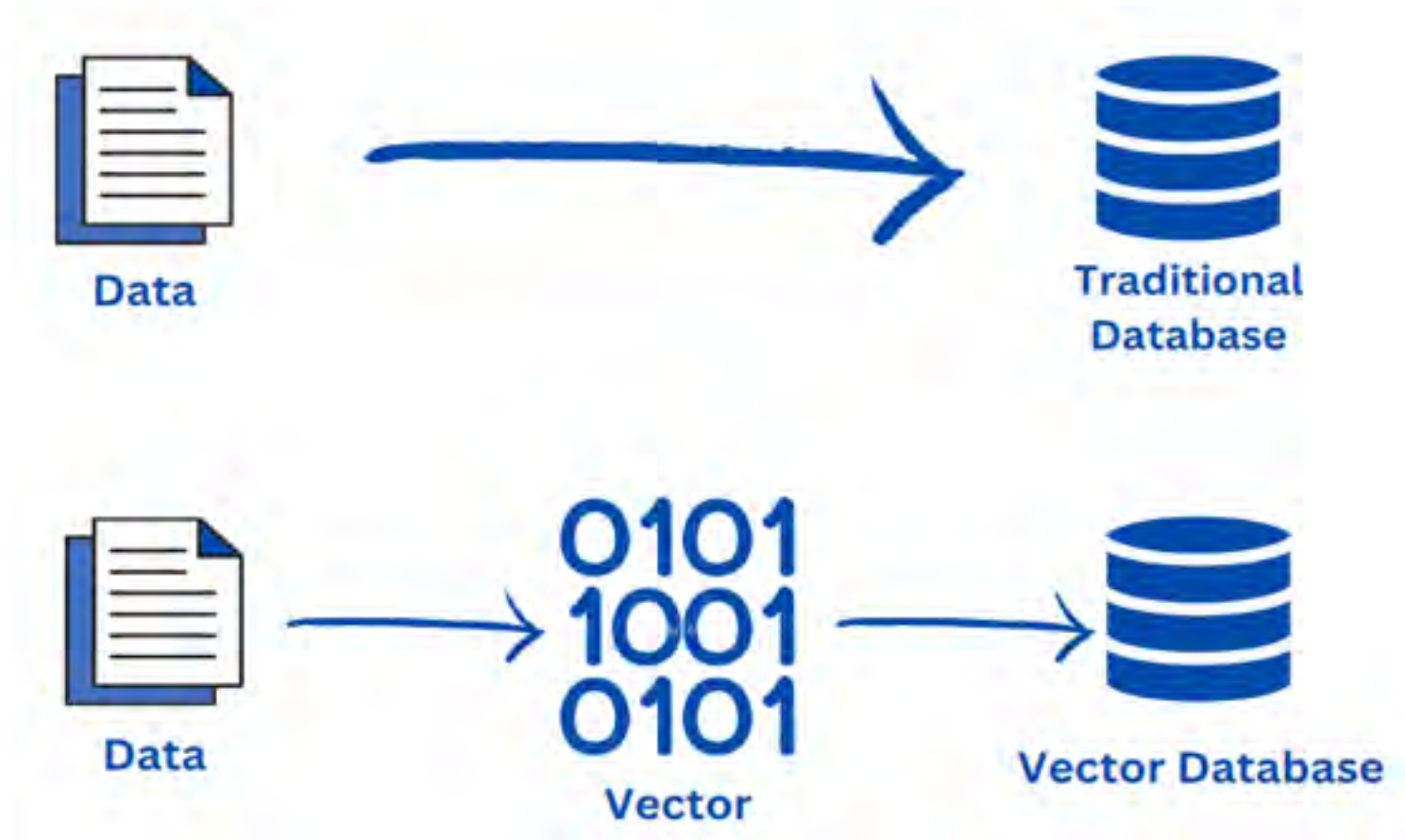


LLM – NAMING CONVENTIONS

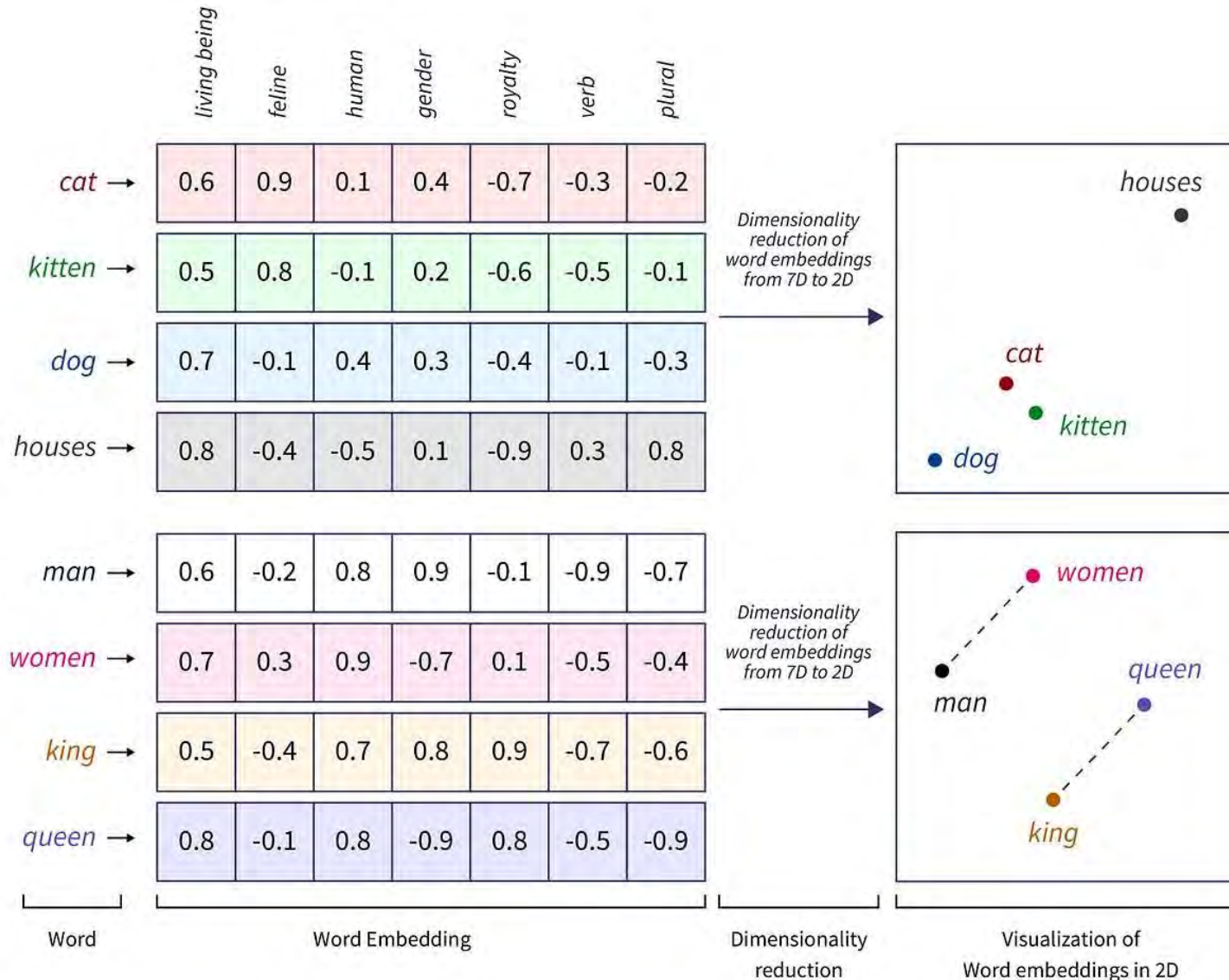
Qwen: Qwen v2.5 Coder 32B (instruct)



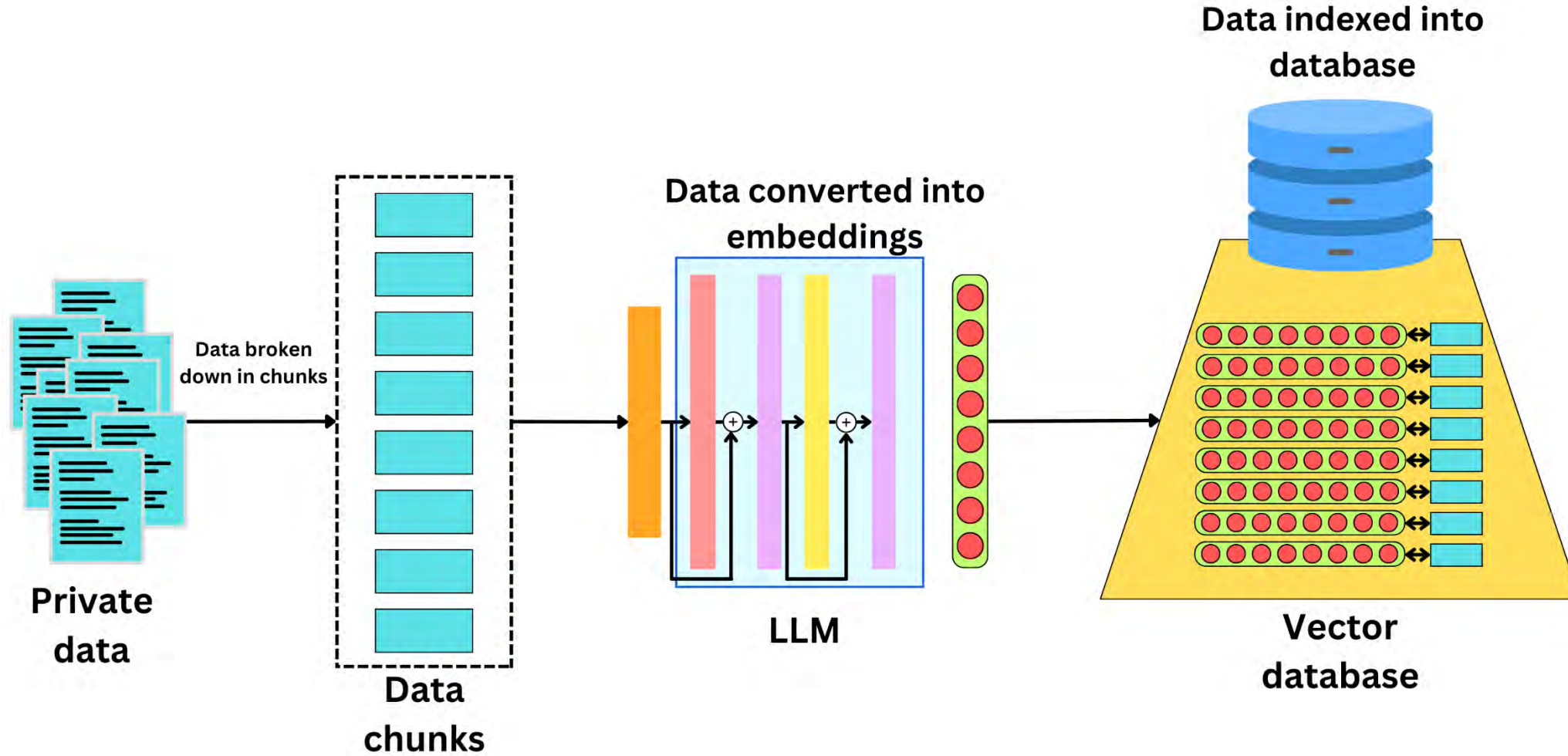
RETRIEVAL AUGMENTED GENERATION - RAG



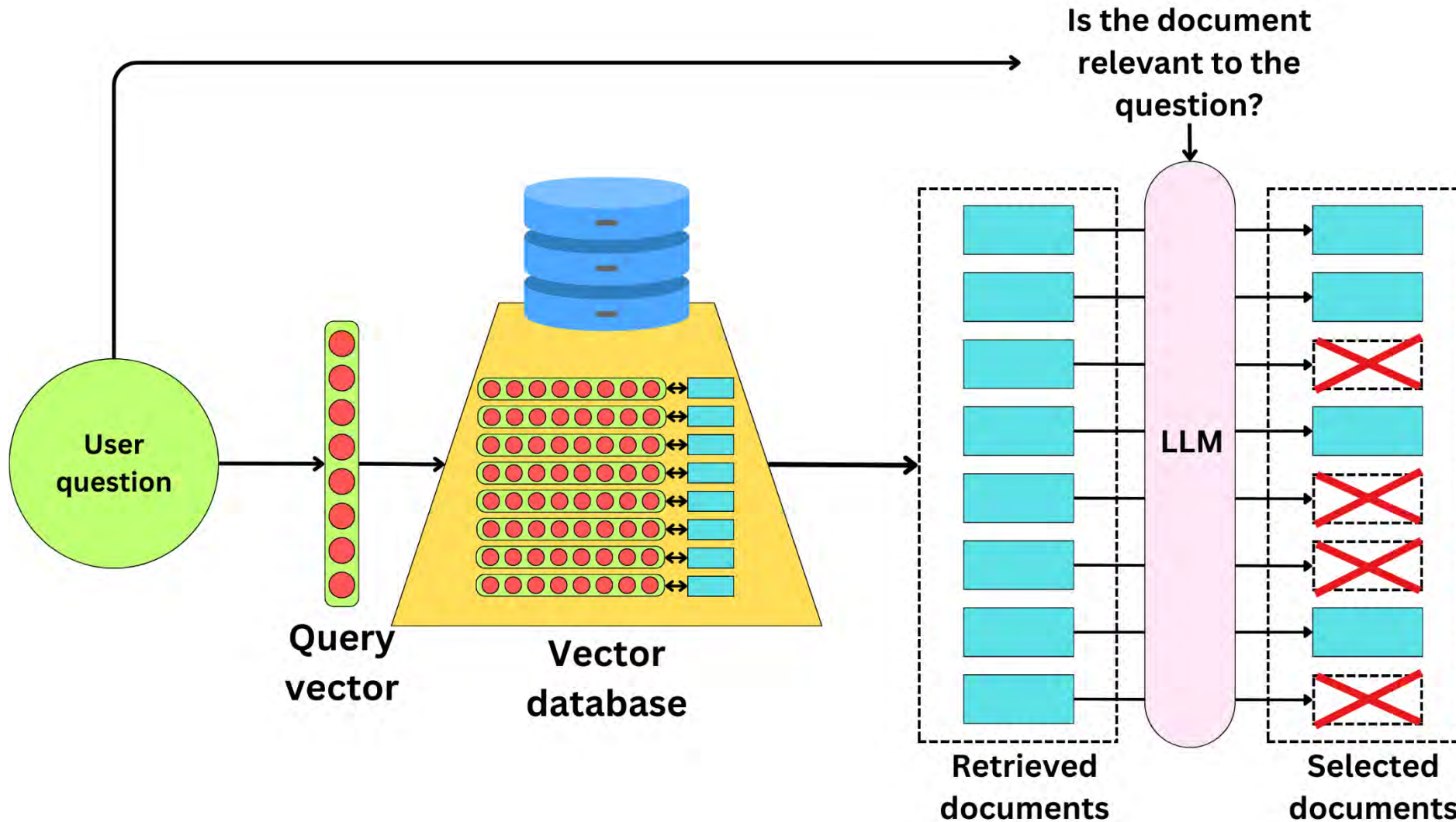
AI EMBEDDINGS



RETRIEVAL AUGMENTED GENERATION - RAG



RETRIEVAL AUGMENTED GENERATION - RAG





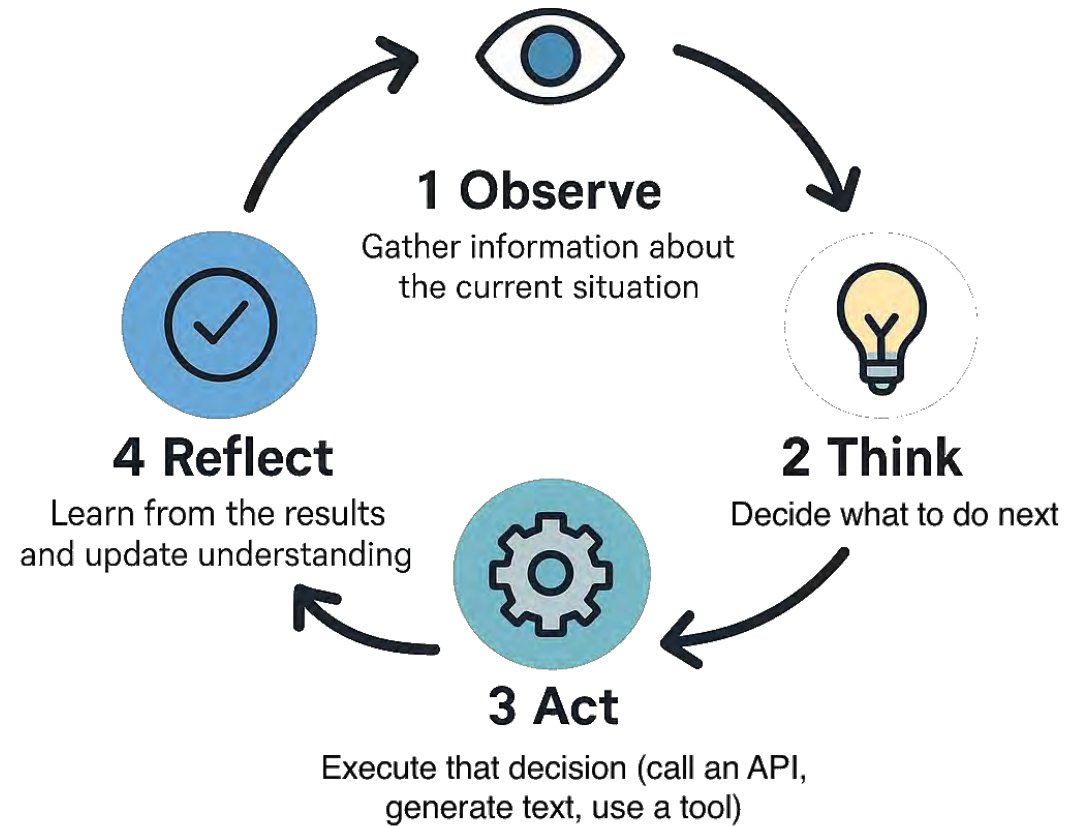
AGENTIC AI

The capacity to act, or the means through which something is achieved.

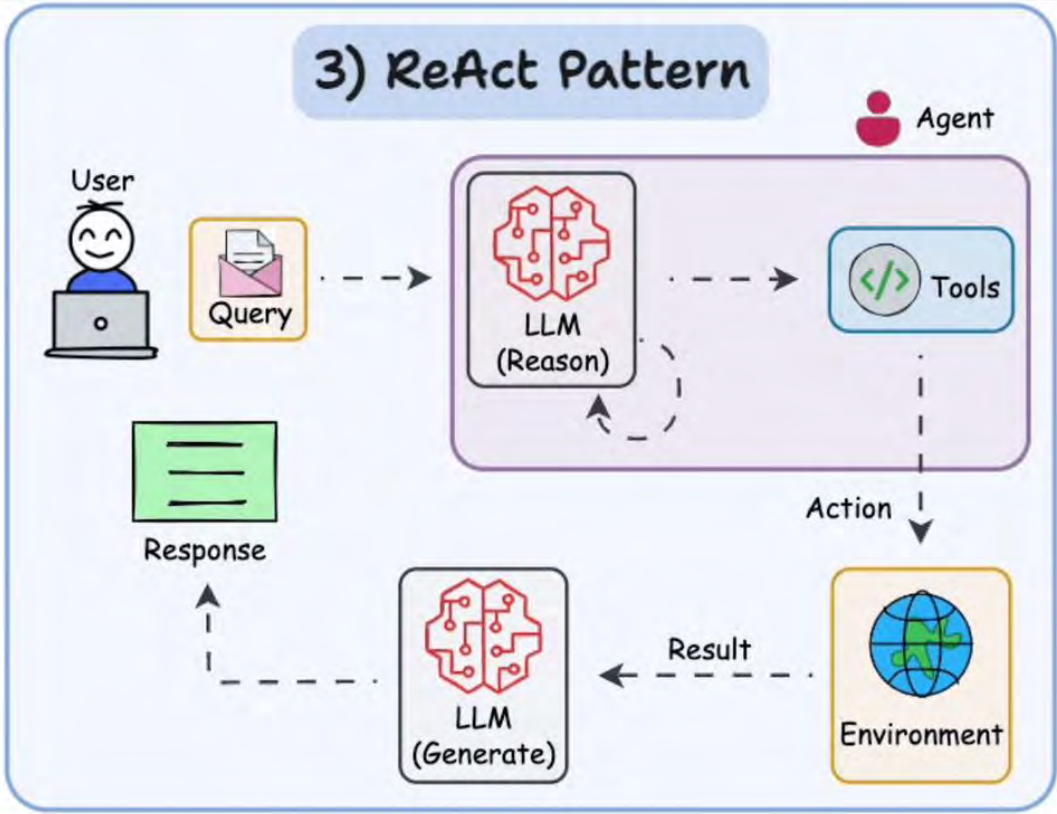
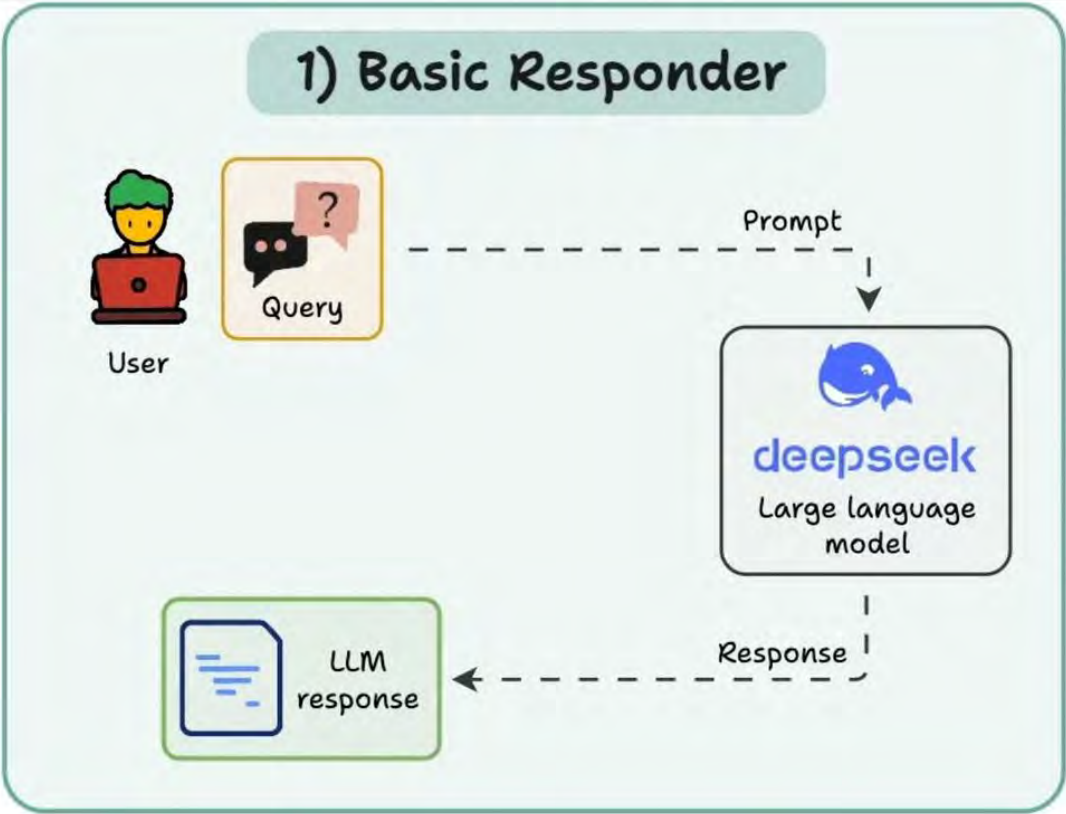
*Example: "Human beings have **agency** to change their circumstances." or "Through the **agency** of a friend, I found a house."*

- Reasoning Models
- Vision Models
- Tool Use Models

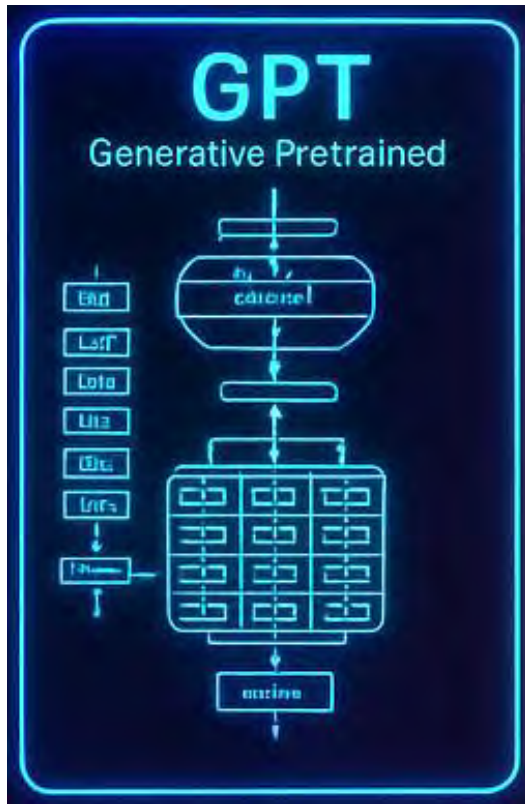
AGENT PATTERN : RE-ACT



AGENT PATTERN : RE-ACT



AGENT LLM TYPES

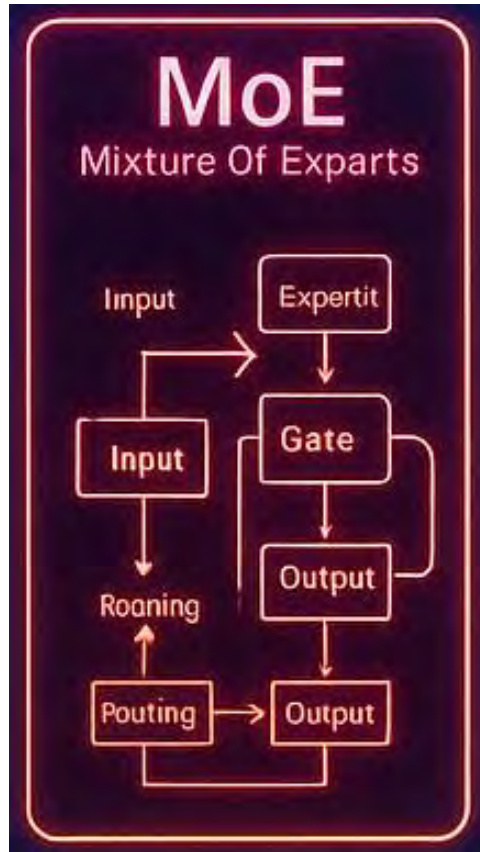


GPT – Generative Pretrained Transformer

The GPT family remains the foundational brain for most agentic systems. These models excel at natural language understanding and generation.

Example: J.P. Morgan's COiN platform leverages GPT-like architectures to parse 12,000 commercial loan agreements in seconds, extracting key clauses and risk indicators. ROI impact: 39% of executives reported measurable productivity ROI from individual GenAI use cases .

AGENT LLM TYPES



MoE – Mixture of Experts

The Mixture of Experts model introduces modularity and scalability by routing queries to specialised sub-networks (“experts”). This design underpins systems such as Gemini 1.5 Pro and Mixtral 8x7B.

Financial example: Tier-1 insurers now use MoE architectures for multi-domain tasks — underwriting, fraud detection, and claims triage — where each expert module is fine-tuned for a domain-specific dataset. Benefit: Lower compute costs and improved specialisation without exponential model growth.

AGENT LLM TYPES



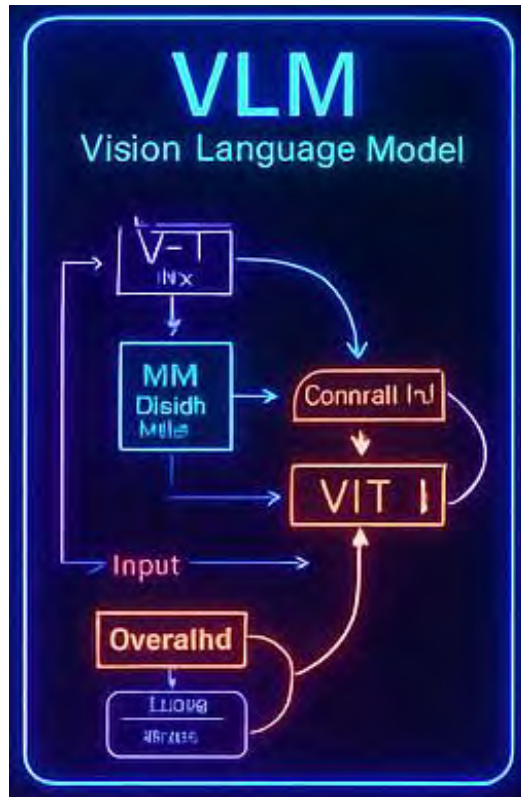
LRM – Large Reasoning Model

The next frontier of intelligence lies in reasoning. LRMs integrate symbolic reasoning, retrieval-augmented generation (RAG), and chain-of-thought planning.

They represent the first generation of “thinking” models—vital for AI agents tasked with multi-step decision-making.

Financial use case: Investment banks deploy LRMs to generate explainable credit decisions and automate structured product documentation. Research reference: DeepMind’s AlphaGeometry and Anthropic’s Claude 3.5 Opus both demonstrate emergent reasoning capacities that mimic human deliberation.

AGENT LLM TYPES



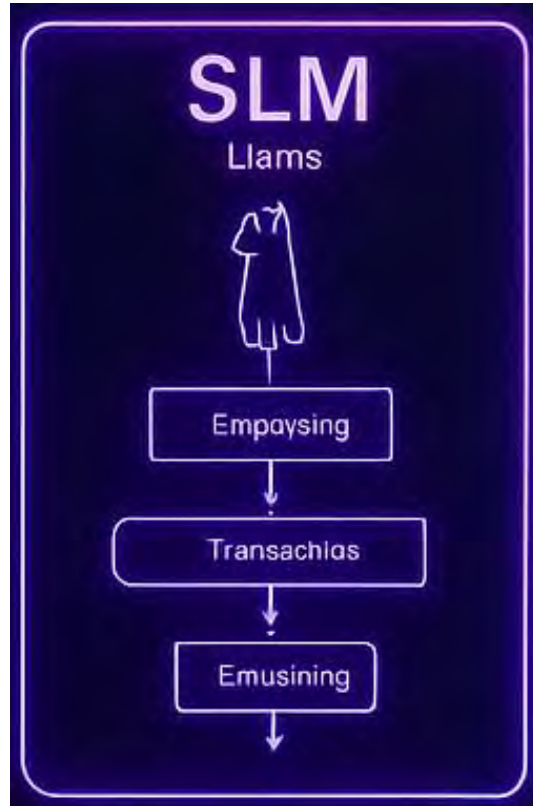
VLM – Vision Language Model

VLMs unify vision and text, allowing agents to see, read, and interpret.

In asset management and insurance, these models power document understanding agents—automatically reading KYC documents, financial statements, and scanned PDFs.

Example: HSBC’s AI agent uses multimodal reasoning to flag anomalies in trade finance documents, combining OCR, NLP, and policy reasoning layers. Frameworks: OpenAI’s GPT-4V and Qwen-VL show how multimodal grounding is becoming a baseline for financial compliance tools.

AGENT LLM TYPES



SLM – Small Language Model

SLMs are lightweight, private, and deployable on-premises or at the edge. They represent the “trusted micro-agents” operating within secured financial environments.

Use case: A Singaporean retail bank runs SLMs to handle low-risk, repetitive internal queries (e.g., HR, policy lookup) entirely offline, preserving data sovereignty.

Trend: As data privacy remains the top concern for 37% of enterprises, SLMs offer a pragmatic alternative to cloud LLMs for regulated sectors.

AGENT LLM TYPES



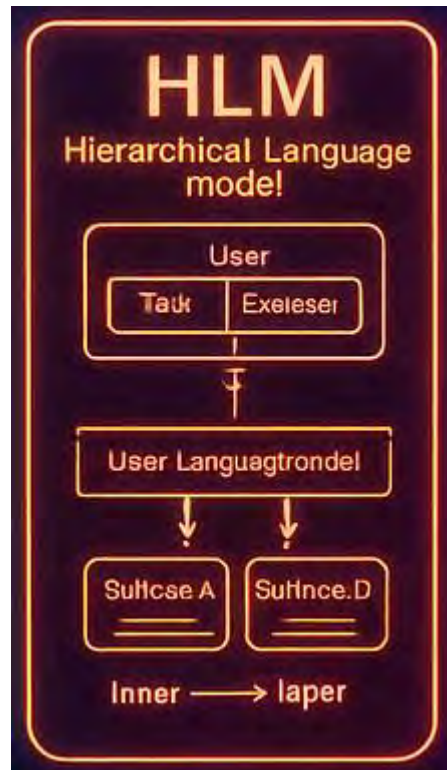
LAM – Large Action Model

LAMs bridge the cognitive-to-actuation gap — enabling AI to not only reason but also perform actions via APIs and tools.

These models sit at the heart of agentic orchestration frameworks like AutoGPT, AgentOS, and LangGraph.

Financial use case: A trading operations agent using a LAM can retrieve data, analyze portfolio exposure, and automatically execute hedging actions — under human approval. Industry example: Bloomberg’s internal LAM prototype integrates task execution pipelines within their terminal ecosystem.

AGENT LLM TYPES



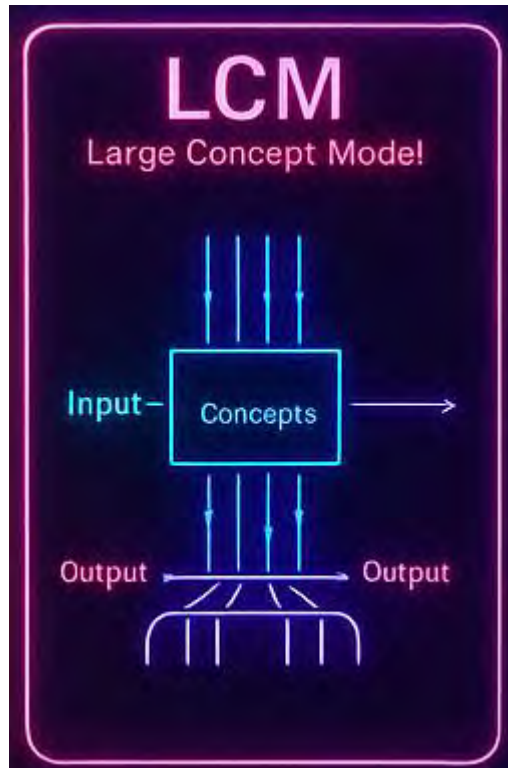
HLM – Hierarchical Language Model

HLMs embody multi-layered cognition, mirroring human organizations.

They orchestrate manager and worker agents, delegating subtasks to specialized sub-agents.

Example: In anti-money-laundering (AML) investigations, an HLM supervises sub-agents: one for transaction graph analysis, one for entity risk scoring, and one for drafting regulatory reports. Research roots: Stanford's CAMEL Framework and Microsoft's AutoGen show the early emergence of hierarchical multi-agent coordination.

AGENT LLM TYPES



LCM – Large Concept Model

LCMs represent the semantic layer of abstraction—models that learn concepts instead of just tokens.

This is critical for aligning AI reasoning with human values, ethics, and strategic foresight—the foundation of what I call Trust-Based Transformation.

In finance: LCMs could serve as governance or ethics co-pilots, translating regulatory intentions (e.g. AI governance frameworks) into operational constraints for AI agents. Philosophical alignment: Fei-Fei Li’s work on human-centered AI and Geoffrey Hinton’s “concept neurons” research underpin this transition toward interpretable concept learning.

AI IN SOFTWARE DEVELOPMENT

The Good:

- Computer Code is Text
- The Code can be predicted with a high degree of accuracy if the model is trained properly.
- Code is less nuanced than a native language, making it a bit more predictable.

The Bad:

- The AI Model can only work with the context that it has. This is usually less than the entire code base.
- The AI does not understand architecture when it writes code – only the current (limited) context.
- AI Code can get itself into a “fix-loop” if the operator is not experienced enough to author the proper input context.
- The AI coder is like a PhD. Level Junior Developer – not a Senior level software engineer.

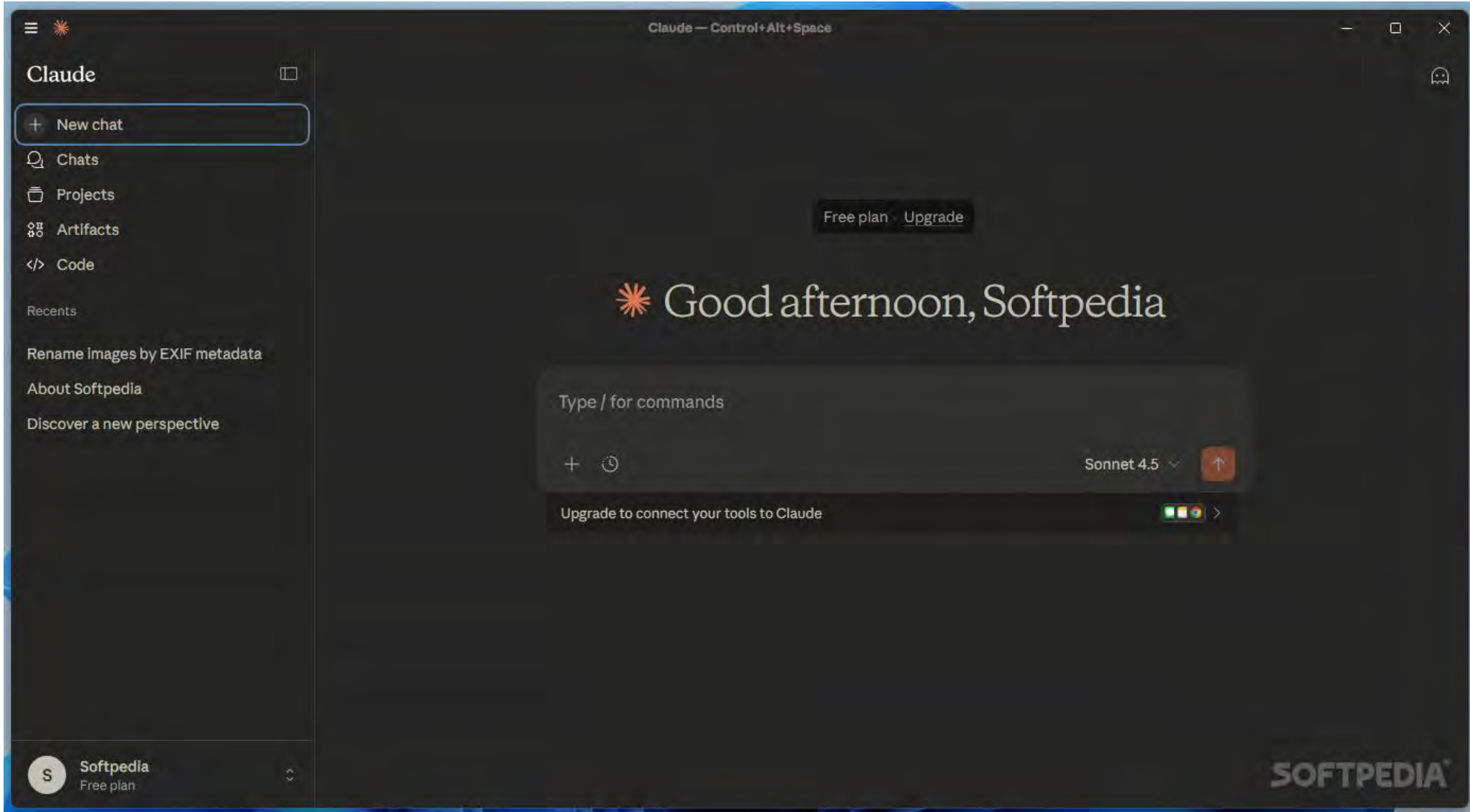


AI IMPLEMENTATIONS

 Claude

 GitHub Copilot

Claude



The screenshot shows the Claude web interface in a dark theme. The window title is "Claude — Control+Alt+Space". On the left, a sidebar contains navigation options: "Claude" (with a sub-menu icon), "New chat" (highlighted with a blue border), "Chats", "Projects", "Artifacts", and "Code". Below these are "Recents" and a list of recent items: "Rename Images by EXIF metadata", "About Softpedia", and "Discover a new perspective". At the bottom left of the sidebar is a profile card for "Softpedia Free plan". The main content area features a "Free plan · Upgrade" button at the top. Below it is a large greeting: "Good afternoon, Softpedia" with the Claude logo. A central input field contains the placeholder text "Type / for commands". To the right of the input field, it shows "Sonnet 4.5" with a dropdown arrow and an upward arrow icon. At the bottom of the input area, there is a message: "Upgrade to connect your tools to Claude" with a small icon of a computer monitor and a right-pointing arrow. The "SOFTPEDIA" logo is visible in the bottom right corner of the interface.

← All projects

Competitive Analysis



I want you to help me do competitive research.

How can I help you today?



Sonnet 4.5



Start a chat to keep conversations organized and re-use project knowledge.

Instructions

I'd like help conducting competitive research.



Files



1% of project capacity used

ElevenLabs
Competitive
Analysis.md

268 lines

MD

Claude

About Us Page / High-Speed Business Scanner for Document Transformation

AM Here's our basic product description for the XC-1000 Scanner:

"High-speed desktop scanner for business use."

Using our technical specifications document and maintaining our brand voice, please create a detailed product description suitable for our website, including key features, benefits, and technical specifications.

XC-1000 Scanner Product Description
Click to open document

I've created a comprehensive product description that aligns with TechSphere Innovations' brand voice and values, emphasizing innovation, quality, and customer success. The description maintains a professional yet accessible tone while highlighting the technical excellence and practical benefits of the XC-1000 Scanner.

Would you like me to adjust any particular aspects of the description? For example, I could:

- Expand on specific use cases
- Add more technical details
- Modify the tone or structure
- Include additional comparison points with similar products

Copy Retry

Claude can make mistakes. Please double-check responses.

AM Reply to Claude...

Claude 3.5 Sonnet (New)

XC-1000 Scanner Product Description

XC-1000 Business Scanner

Transform your document management with enterprise-grade scanning power

The XC-1000 Scanner embodies TechSphere Innovations' commitment to excellence, delivering a powerful desktop scanning solution that simplifies complex document management challenges. Designed for modern businesses that demand both speed and precision, the XC-1000 seamlessly bridges the gap between physical and digital workflows.

Overview

Transform your office efficiency with our state-of-the-art desktop scanner, engineered to handle high-volume scanning needs while maintaining exceptional quality. The XC-1000 combines innovative technology with user-friendly design, making it the ideal choice for businesses looking to streamline their document digitization process.

Key Features

Exceptional Speed & Quality

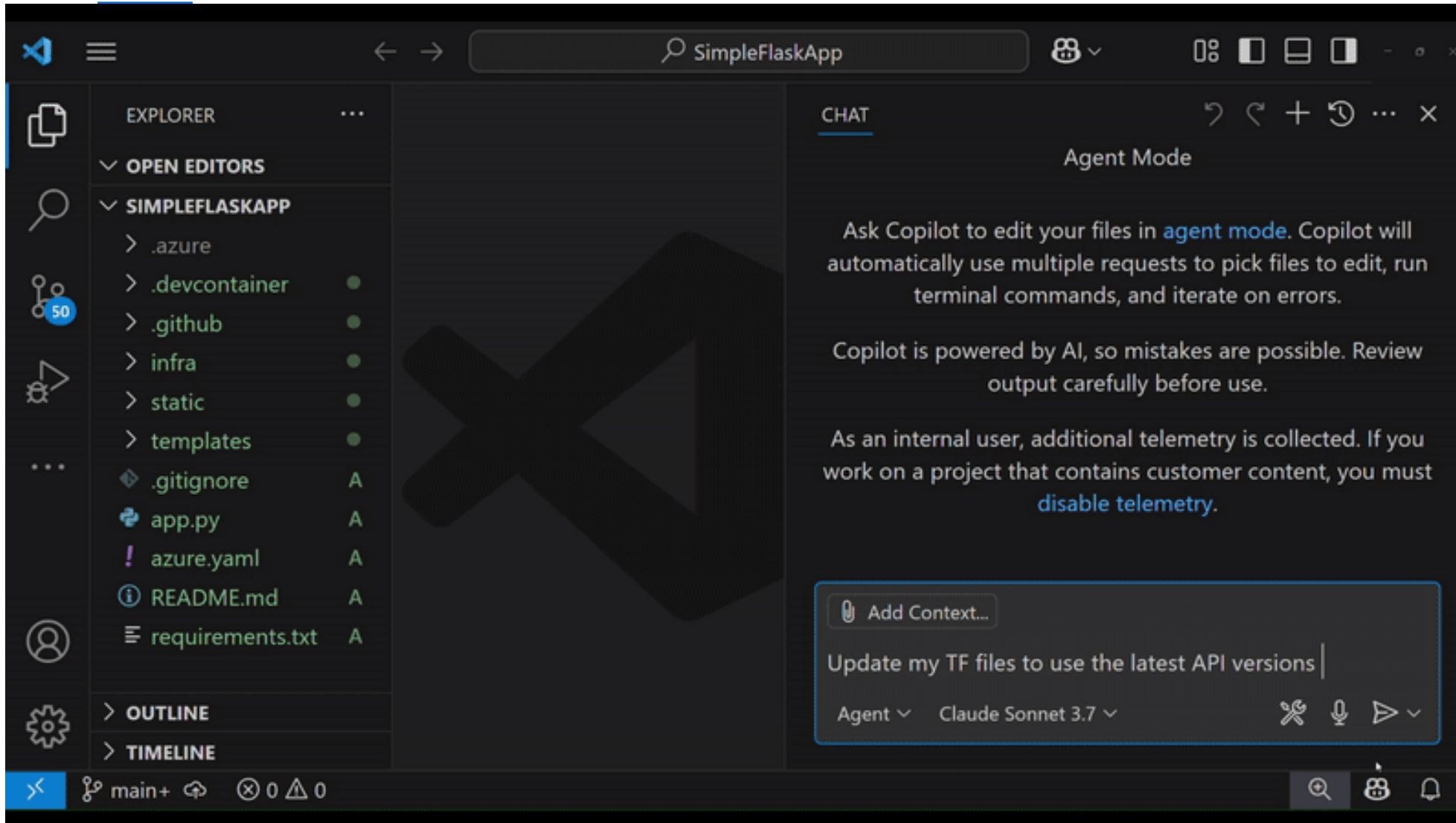
- Lightning-fast scanning at 60 pages per minute
- Crystal-clear resolution up to 600 dpi
- Advanced image enhancement technology
- Dual-sided scanning capability

Last edited just now

Publish

GitHub Copilot

Chat - Agent



The screenshot shows the Visual Studio Code interface with the GitHub Copilot Chat Agent Mode. The Explorer sidebar on the left displays the file structure for the project 'SIMPLEFLASKAPP', including folders like '.azure', '.devcontainer', '.github', 'infra', 'static', and 'templates', and files like '.gitignore', 'app.py', 'azure.yaml', 'README.md', and 'requirements.txt'. The Chat panel on the right is titled 'CHAT' and 'Agent Mode'. It contains the following text:

Ask Copilot to edit your files in **agent mode**. Copilot will automatically use multiple requests to pick files to edit, run terminal commands, and iterate on errors.

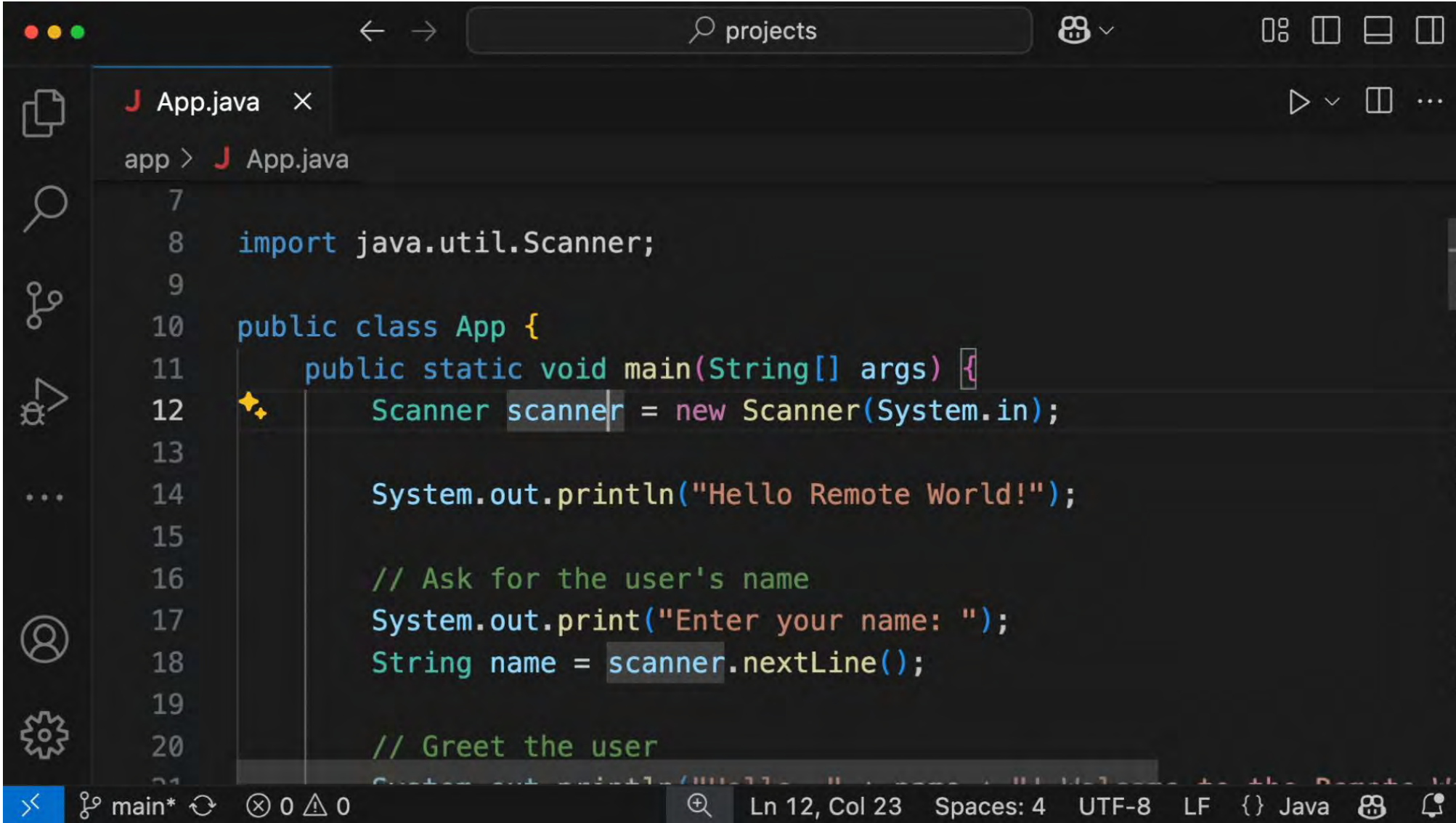
Copilot is powered by AI, so mistakes are possible. Review output carefully before use.

As an internal user, additional telemetry is collected. If you work on a project that contains customer content, you must **disable telemetry**.

The chat input field contains the prompt: "Update my TF files to use the latest API versions". Below the input field, the model is identified as "Agent" and "Claude Sonnet 3.7".

GitHub Copilot

Autocomplete



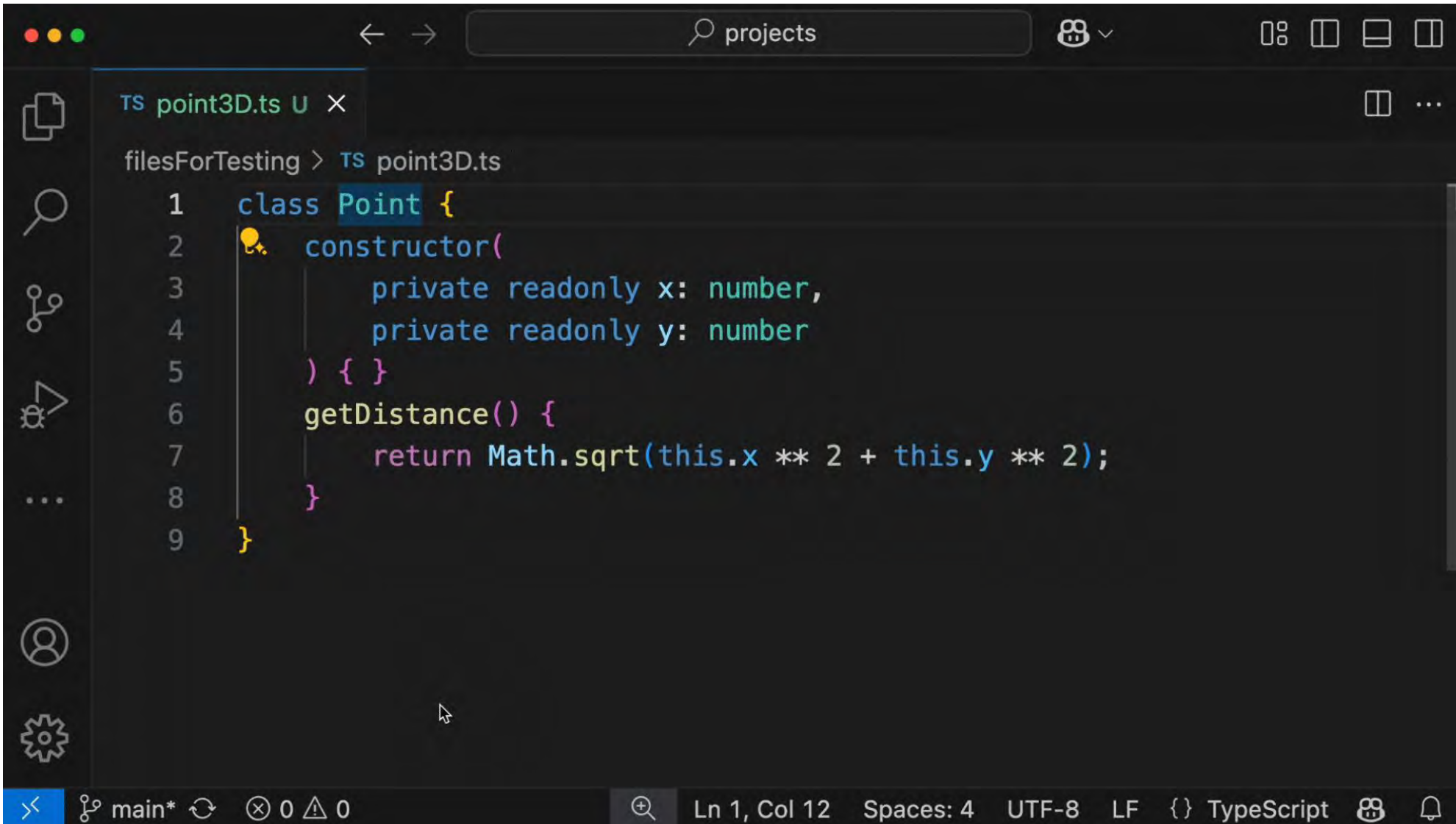
The screenshot shows an IDE window with a dark theme. The top bar includes navigation arrows, a search bar containing 'projects', and window management icons. The file explorer on the left shows 'App.java' selected. The main editor displays the following Java code:

```
7
8 import java.util.Scanner;
9
10 public class App {
11     public static void main(String[] args) {
12         Scanner scanner = new Scanner(System.in);
13
14         System.out.println("Hello Remote World!");
15
16         // Ask for the user's name
17         System.out.print("Enter your name: ");
18         String name = scanner.nextLine();
19
20         // Greet the user
```

At line 12, the word 'Scanner' is highlighted, and a tooltip shows a yellow star icon and the text 'Scanner scanner = new Scanner(System.in);'. The status bar at the bottom indicates 'main*', '0' errors, '0' warnings, 'Ln 12, Col 23', 'Spaces: 4', 'UTF-8', 'LF', and 'Java'.

GitHub Copilot

Intent Inference

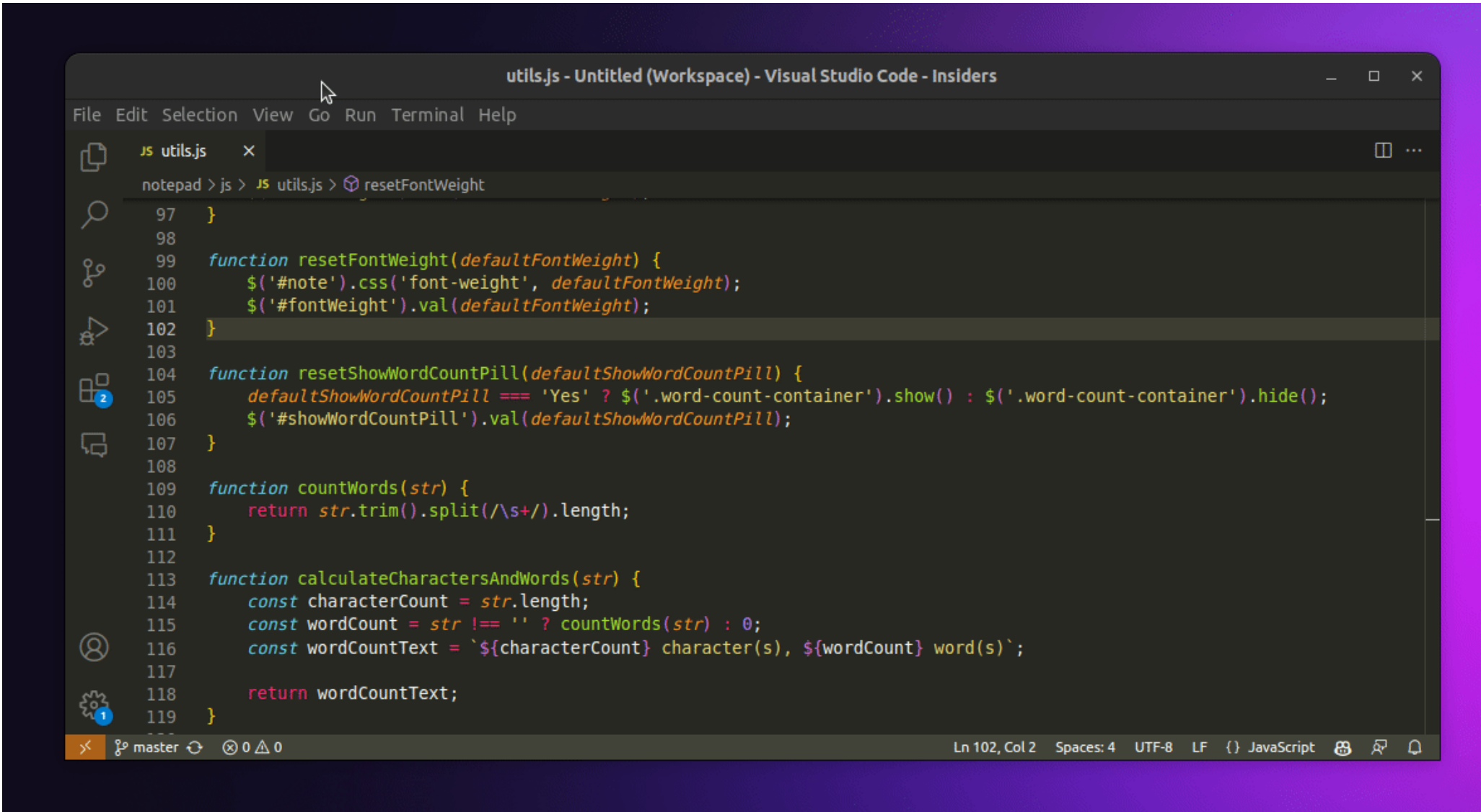


```
TS point3D.ts U x
filesForTesting > TS point3D.ts
1  class Point {
2      constructor(
3          private readonly x: number,
4          private readonly y: number
5      ) { }
6      getDistance() {
7          return Math.sqrt(this.x ** 2 + this.y ** 2);
8      }
9  }
```

main* 0 0 Ln 1, Col 12 Spaces: 4 UTF-8 LF {} TypeScript

GitHub Copilot

Inline Chat



```
utils.js - Untitled (Workspace) - Visual Studio Code - Insiders
File Edit Selection View Go Run Terminal Help
JS utils.js
notepad > js > JS utils.js > resetFontWeight
97 }
98
99 function resetFontWeight(defaultFontWeight) {
100     $('#note').css('font-weight', defaultFontWeight);
101     $('#fontWeight').val(defaultFontWeight);
102 }
103
104 function resetShowWordCountPill(defaultShowWordCountPill) {
105     defaultShowWordCountPill === 'Yes' ? $('.word-count-container').show() : $('.word-count-container').hide();
106     $('#showWordCountPill').val(defaultShowWordCountPill);
107 }
108
109 function countWords(str) {
110     return str.trim().split(/\s+/).length;
111 }
112
113 function calculateCharactersAndWords(str) {
114     const characterCount = str.length;
115     const wordCount = str !== '' ? countWords(str) : 0;
116     const wordCountText = `${characterCount} character(s), ${wordCount} word(s)`;
117
118     return wordCountText;
119 }
```

Ln 102, Col 2 Spaces: 4 UTF-8 LF {} JavaScript

VIBE CODING

The Concept:

- A Non-Coder with BA experience can create an app just by talking to the AI.

The Reality:

- Vibe Coding works very well for a small prototype, a conceptual demo or a one-off project.
- Vibe Coding can wreck havoc long-term.

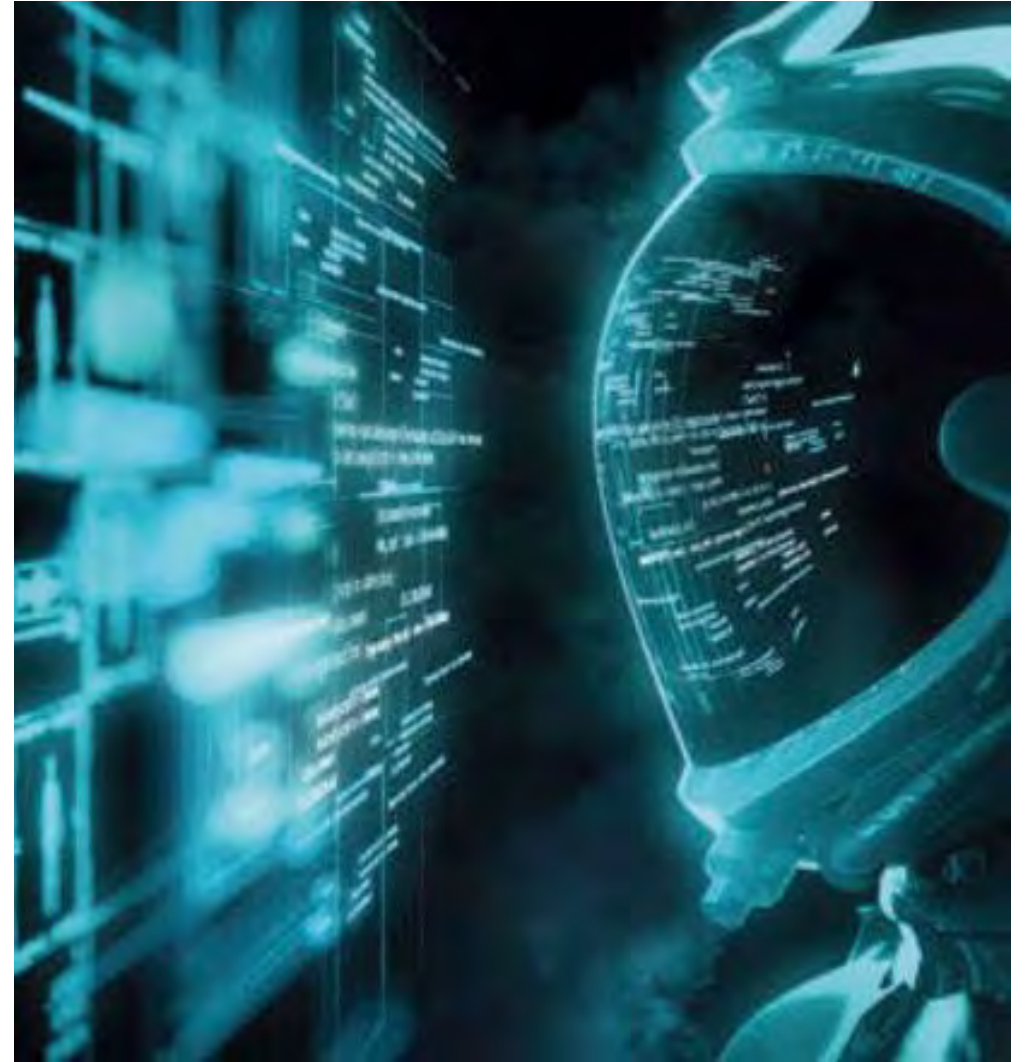
The Reason:

- AI is not good at making meaningful decisions. It will always take the path of least resistance.
- AI loses context as the code grows.
- AI does NOT consider infrastructure costs, user load, security, data integrity, etc. It does what it is told.
- Only a seasoned professional can craft a long-term solution that works for the interest of the business.



AI INTO THE FUTURE

- AI will become more and more 'agentic'.
- LLMs will get smaller and more specialized.
- AI will become more localized over time.
- AI will become systematized, infrastructure
- Smart-Tech will get 'smarter'.
- Robotics will take a quantum leap forward.



ROI – HOW YOU CAN LEVERAGE AI

HOPE / Durability

- **Build Systems**, not just Chat Bots
- **Use specialized models**, where appropriate
- Leverage **workflow automation** and **background infrastructure**
- **Hybrid use of local and cloud models**, for added security and cost savings.
- Local **Data Storage** for **AI Workflows** - (Vector DBs, Data Warehouse, etc.)
- Narrow, High-value **workflow automation**

HYPE / "Cooling off"

- **AI will replace people**
- **One Model**, can do everything!
- **Bigger Models** are **Better!**
- **Chat-bot first** thinking
- **Benchmark Theater** - (not what business needs)

HOW CAN THEORIS HELP?

- AI Use Training
- AI Assessments
- AI Project/Workflow Consulting
- Development Team AI Tooling



