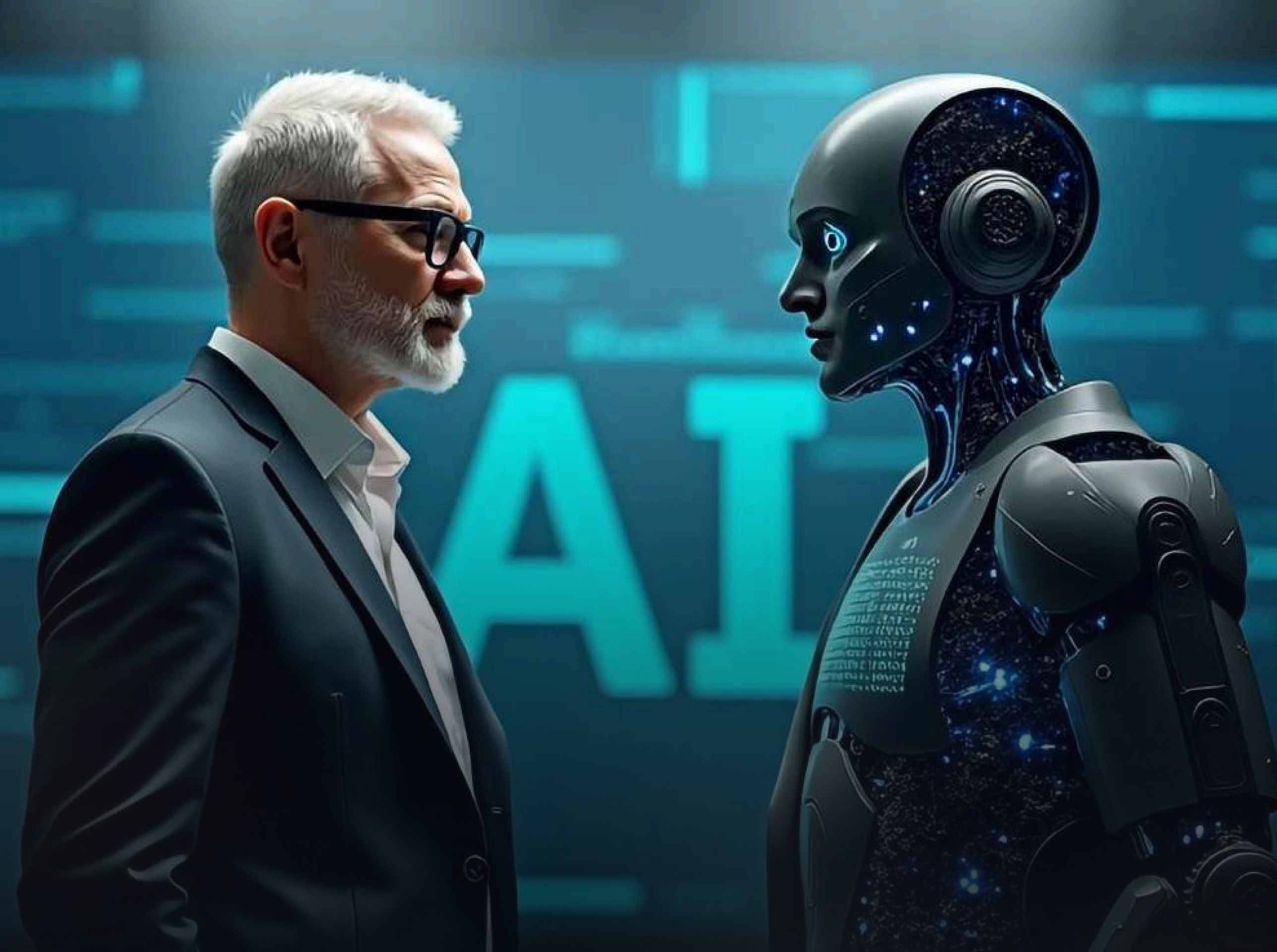


WHITEPAPER

AI AGENTS

The Dawn of a New Workforce

How Enterprises are Delegating Cognitive Tasks to AI

Gleecus TECHLABS INC

Forward Thinking Digital Innovation

Table Of Contents

Executive Summary

Nvidia CEO Jensen Huang proclaimed at CES 2025 that AI Agents represent a multi-trillion dollar industry. This white paper aims at demystifying what AI Agents are and the underlying technologies behind them. Labelled as “AI’s killer function” by OpenAI CEO Sam Altman AI Agents takes things beyond generative modelling and are expected to give rise to a new workforce that operate alongside humans like a fleet of interns, performing repetitive tasks and improving overall productivity.

Today, apparently 56% of workers use generative AI on the job as researched by The Conference Board. In this AI-enabled stage of business it is essential to be prepared about the transformative potential of AI agents, and the level of automation they can offer with the existing AI tech horizon. Reading about the automation capabilities of AI Agent may seem like the future but the future is already here considering the rapid growth in Generative AI technologies.

Unlike simple bots AI Agents can handle entire processes, making decisions and adjusting their actions based on real-time data and changing circumstances. A report from McKinsey states that “about half of the activities (not jobs) carried out by workers could be automated,” with AI agents playing a crucial role in this transformation.

This distinction between automating tasks versus entire jobs is critical — it suggests a future where human-AI collaboration becomes the norm, rather than wholesale replacement. As we weave agentic AI capabilities into our businesses, we will likely deconstruct jobs into individual tasks and then identify the tasks that can be fully automated by these new AI technologies and agents.

This white paper takes a deep dive into the frameworks and technologies behind AI Agents. It explores the fundamental logic and reasoning behind how AI Agents perform. Read on to learn more about the different types of AI Agents, their industrial use cases, and potential challenges associated with their adoption.

What are AI Agents?

An AI Agent is a software program that can autonomously execute predetermined goals set by a user or another agent. They leverage complex NLP (Natural Language Processing) technologies to interact with the environment, chat with the user, gather data, and designs workflows to execute the predetermined goals.

An agent takes the power of generative AI a step further, because instead of just assisting you, agents can work alongside you or even on your behalf. Agents can do a range of things, from responding to questions to more complicated or multistep assignments. What sets them apart from a personal assistant is that they can be tailored to have a particular expertise.

> *"AI agents are advanced artificial intelligence systems that are able to complete a task or make a decision,"*
>
> **- Adnan Ijaz, director of product management, Amazon Q Developer**

Agents are like layers on top of the language models that observe and collect information, provide input to the model and together generate an action plan and communicate that to the user — or even act on their own, if permitted. Both agents and models are equally important pieces of the puzzle, as far as generative AI (GenAI) tools go.

Some agents can be seen in the real world—as robots, automated drones, or self-driving cars. Others are purely software-based, running inside computers to complete tasks. The actual aspect, components, and interface of each AI agent vary widely depending on the task it's meant to work on.

And unlike with a GenAI chatbot, you don't need to constantly send prompts with new instructions. AI agents will run once you give them an objective or a stimulus to trigger their behavior. Depending on the complexity of the agent system, it will use its processors to consider the problem, understand the best way to solve it, and then take action to close the gap to the goal. While you may define rules to have it gather your feedback and additional instructions at certain points, it can work by itself.

There's a bit of confusion between AI agents and regular agent software. The latter falls under robotic process automation (RPA). These apps can use a computer like a human user, looking at screens, clicking elements, and automating work. They're based on pre-determined rules and deal with structured data, lacking flexibility and adaptability. They don't use AI at all—but you can definitely integrate AI into it to give it extra powers.

How Does an AI Agent Work?

AI agents in general follows a three steps approach to execute their assigned tasks:

Determine goals

The AI Agent receives a specific set of instructions or goal from the user. The user also defines the environment and establishes available tools. These goals define the specific objectives that the agent must achieve, guiding its behavior throughout the process. The goals can vary in complexity, from simple, single-step tasks to more intricate, multi-step processes that require careful coordination and planning.

Given the user's goals and the agent's available tools, AI Agent breaks down the goal into smaller actionable tasks called subtasks.

This process is critical because it allows the agent to systematically address each part of the problem, ensuring that all necessary steps are covered to reach the final objective. For example, if an AI agent is tasked with planning a large-scale event, it might decompose the goal into subtasks such as selecting a venue, coordinating vendors, sending invitations, and managing guest lists. Each of these subtasks is then tackled individually, with the broader goal in mind.

Task decomposition is not just about breaking down tasks; it also involves sequencing these subtasks in an order that optimizes the agent's efficiency. The agent considers various factors, such as dependencies between tasks, time constraints, and resource availability, to create a detailed action plan.

Acquire information

When an AI agent encounters a task that requires information beyond its current knowledge base, it turns to external data sources. At the core of AI Agents are LLMs and orchestrators that help with tool calling on the backend to obtain up-to-date information. This involves extracting conversation logs, interaction with other agents or machine learning models, connecting to an external database, or searching up the internet.

Once the data is collected, AI agents utilize algorithms to retrieve specific information relevant to the task at hand. This can involve:

Search Algorithms: Employing keyword searches or semantic searches to find pertinent information across databases or the internet.

Natural Language Processing: Analyzing text to extract meaningful insights or understand user queries in conversational AI applications.

After each interaction with external tools, the agent reassesses its plan, making necessary adjustments to ensure that it stays on track toward achieving the goal. This iterative reasoning is particularly important in dynamic environments where conditions can change rapidly.

Implement tasks

With sufficient data gathered, the AI Agent methodically implements the task at hand.

To execute tasks effectively, AI agents need to understand the context surrounding the information they gather. This involves:

Contextual Analysis: Using machine learning models to determine the relevance of the information based on the specific task requirements and user intent.

Knowledge Graphs: Leveraging structured representations of knowledge that illustrate relationships between different concepts, helping agents understand context better.

AI Agents utilize decision-making frameworks to determine the best course of action. This can involve:

Rule-Based Systems: Following predefined rules and logic to make decisions based on specific criteria.

Machine Learning Models: Using trained models to predict outcomes and suggest actions based on historical data.

Once it accomplishes a task, the agent removes it from the list and proceeds to the next one. In between task completions, the agent evaluates if it has achieved the designated goal. For this they seek feedback from human user and any other external agent involved. Feedback assists in iterative refinement of AI Agents by allowing them to adjust to user preferences for future goals and avoid repeating the same mistakes.

Learning and reflection are integral components of an AI agent's operation, enabling it to improve over time and adapt to new challenges.

Learning and reflection

AI agents continuously learn from each interaction, refining their algorithms to improve accuracy and effectiveness. They update their knowledge base and use feedback to enhance future interactions. This continuous learning capability ensures that AI agents remain effective and relevant, even as customer expectations and business environments change.

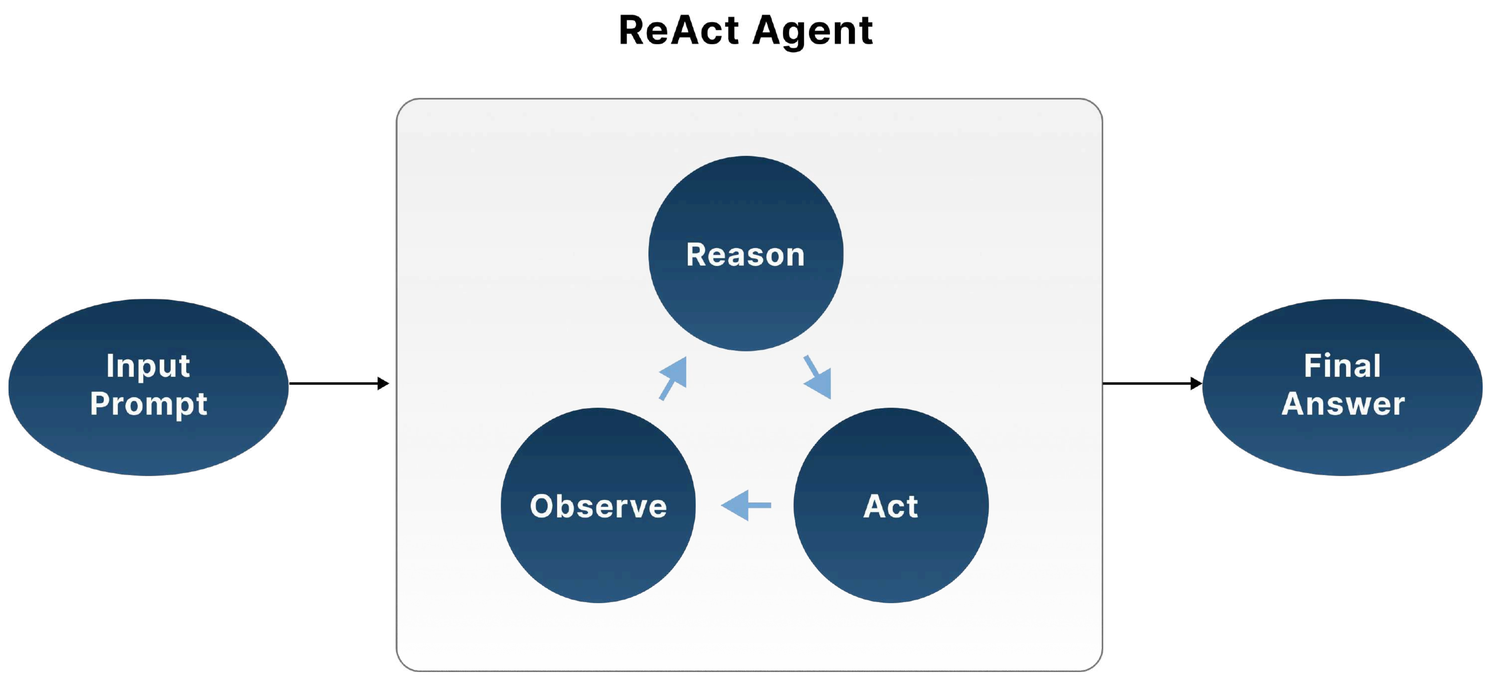
Reasoning Paradigm of AI Agent

There is not one standard architecture for building AI agents. Several paradigms exist for solving multi-step problems.

ReAct (Reasoning and Action)

With this paradigm, we can instruct agents to "think" and plan after each action taken and with each tool response to decide which tool to use next. These Think-Act-Observe loops are used to solve problems step by step and iteratively improve upon responses.

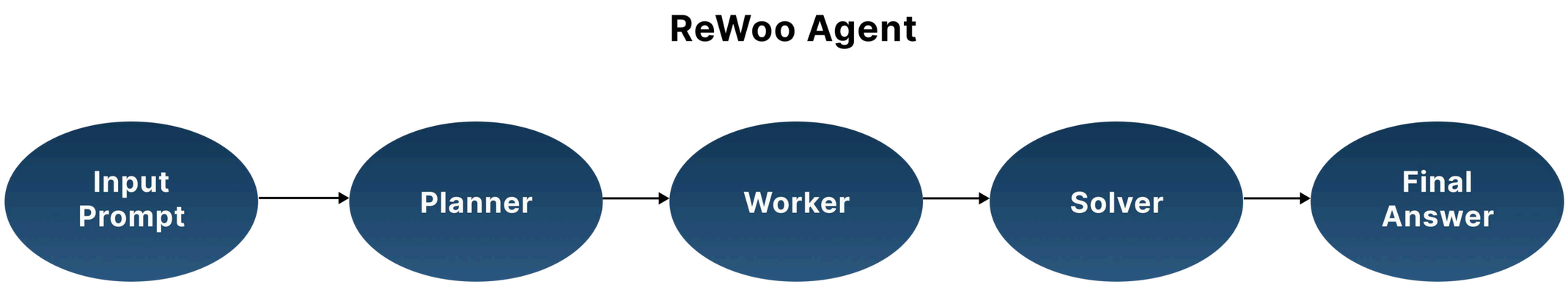
Through the prompt structure, agents can be instructed to reason slowly and to display each "thought". The agent's verbal reasoning gives insight into how responses are formulated. In this framework, agents continuously update their context with new reasoning. This can be interpreted as a form of Chain-of-Thought prompting.



ReWOO (Reasoning WithOut Observation)

The ReWOO method, unlike ReAct, eliminates the dependence on tool outputs for action planning. Instead, agents plan upfront. Redundant tool usage is avoided by anticipating which tools to use upon receiving the initial prompt from the user. This is desirable from a human-centered perspective since the user can confirm the plan before it is executed.

The ReWOO workflow is made up of three modules. In the planning module, the agent anticipates its next steps given a user's prompt. The next stage entails collecting the outputs produced by calling these tools. Lastly, the agent pairs the initial plan with the tool outputs to formulate a response. This planning ahead can greatly reduce token usage and computational complexity as well as the repercussions of intermediate tool failure.



Benefits of AI Agent

Task automation

Automating tasks optimizes workflows and enhances productivity. AI Agents can perform complex tasks without human intervention. They can automatically gather information, find solutions, and make decisions. This leaves room for enterprises to divert their human resources into mission-critical or creative activities deriving greater value. Also, automation with AI Agents facilitates fast scaling and provides freedom from manual errors.

Improved customer experience

AI Agents are agentic chatbots that use conversational AI to interact with humans. They have the necessary tools, memory, and analytics capabilities to analyze human sentiment. This enables them to generate responses that are more comprehensive, accurate and personalized to the user. The feedback-based improvement that is integral to an AI Agent helps them to self-correct and adapt them to meet user expectations yielding higher customer satisfaction over time.

Greater performance

Multi-agent frameworks empower one agent to gather knowledge from another specialized agent. This makes them capable of serving complex goals consisting of diverse subtasks. This backend collaboration of AI Agents and the ability to fill information gaps are unique to agentic frameworks, making them a high-performance solution and a meaningful advancement in artificial intelligence.

24/7 availability

Unlike human workers, AI agents can operate continuously without breaks or downtime. This constant availability is particularly beneficial in industries such as customer support, where immediate assistance can significantly enhance user satisfaction.

Personalization

AI agents can tailor experiences based on individual user preferences and behaviors. For example, recommendation systems powered by AI analyze past interactions to suggest products or content that align with users' interests, enhancing engagement and satisfaction.

Scalability

AI agents can handle an increasing volume of tasks without a proportional increase in costs or resources. This scalability allows businesses to grow and adapt to changing demands efficiently.

Cost Savings

By automating tasks and improving efficiency, AI agents can lead to significant cost savings for organizations. Reduced labor costs and increased productivity contribute to a healthier bottom line.

Continuous learning and improvement

AI agents are designed to learn from interactions over time, allowing them to adapt and improve their performance continually. This capability ensures that they remain effective as user needs evolve.

Types of AI Agent

Simple reflex agents

Simple reflex agents are the simplest form of agent that works triggering only to a set of predefined conditions. They don’t store data, interact with other agents, and act only on the available current data. The agents depend on a set of predefined rules or conditional action pairs. Such agents are suitable for simple tasks that don’t require extensive training. The agents are only effective in environments that are fully observable, granting access to all necessary information, as they lack internal memory. This simplicity makes them easy to enforce and apprehend, making them perfect for introductory research in Artificial Intelligence. For example, a Captcha system validates a human user based on the images he selects from a set of images or the letters he types in.

Model-based reflex agents

Model-based reflex agents are an advanced form of simple reflex agents. They store a model of their world around in an internal memory. Based on current data and stored information they evaluate the probable outcome and consequences before acting. They can make decisions on their own instead of being solely driven by rules. Such agents can function in a partially observable environment. Take the example of a robot vacuum cleaner, it senses obstacles such as furniture and adjusts around them while cleaning the room. It also stores information about the areas wiped to avoid repetitively working on them.

Condition action rule

Model-based reflex agents use condition-action rules to make decisions and act in real-time, based on their perception of the environment. It represents a simple form of logic that dictates how the agent should respond to specific conditions in its environment. Rules can be defined manually or learned through machine learning techniques. These rules or logic specify actions to be taken in response to certain conditions perceived by the agent.

Condition-action rules are often represented in the form of "if-then" statements, where the "if" part specifies the condition and the "then" part specifies the action.

For example:

- If an obstacle is detected in front of the robot, then stop and change direction.
- If the temperature exceeds a certain threshold in a climate control system, then activate the cooling system.
- If the demand for a product exceeds the available inventory, then increase production.

Goal-based agents

Goal-based AI agents represent a sophisticated approach in artificial intelligence (AI), where agents are programmed to achieve specific objectives. These agents are designed to plan, execute, and adjust their actions dynamically to meet predefined goals. This approach is particularly useful in complex environments where flexibility and adaptability are crucial. A best example can be a navigation recommendation system finding the fastest route to your destination after analyzing traffic conditions.

Key Concepts of Goal-Based AI Agents

Goals

Goals are the specific objectives that the agent aims to achieve. These can range from simple tasks, such as sorting objects, to complex missions, such as navigating a robot through a maze, solving a puzzle, or managing resources in a simulated environment. Goals provide a clear direction for the agent's actions and decisions.

Planning

Planning involves determining the sequence of actions required to achieve the goal. This process can be complex, involving predictive models, heuristics, and algorithms to evaluate possible future states and actions. Effective planning allows agents to anticipate potential obstacles and devise strategies to overcome them.

Execution

Execution is the phase where the agent carries out the planned actions. This involves interacting with the environment and performing tasks that bring the agent closer to its goal. Successful execution requires precise coordination of actions and real-time responsiveness to changes in the environment.

Adaptation

Adaptation is essential as the agent interacts with its environment. It may encounter unexpected obstacles or changes, and adaptation involves modifying plans and actions in response to new information, ensuring the agent remains on track to achieve its goal. This ability to adapt makes goal-based agents robust and flexible.

Utility-based agents

Utility theory is a fundamental concept in economics and decision theory. This theory provides a framework for understanding how individuals make choices under uncertainty. Utility-based agents select the sequence of actions that reach the goal with maximum desirable outcome. Utility value, a metric is assigned to each scenario to evaluate the usefulness of an action. This metric can be set based on factors like progression toward the goal, time requirements, or computational complexity. The aim of this agent is not only to achieve the goal but the best possible way to reach the goal. The agent selects the set of actions that sum up the highest utility value or provides maximum rewards to the user. A typical utility function is to earn maximum points in a game. When presented with different possible actions, the utility-based agent chooses the one expected to optimize its utility according to its utility function.

Components of Utility-Based Agents

Utility function

The utility function is a core element of utility-based agents, serving as a mathematical representation of the agent's preferences. It assigns a numerical value (utility) to each possible outcome, reflecting the desirability or satisfaction associated with that outcome.

State space

The state space S is the set of all possible states sss that the agent can occupy. The state space defines the environment in which the agent operates. By understanding all possible states, the agent can better predict the consequences of its actions and plan accordingly.

Actions

The set of actions A consists of all possible moves or decisions the agent can make in any given state. Actions enable the agent to interact with and change its environment. By selecting the appropriate action, the agent aims to move towards states with higher utility.

Transition model

The transition model describes how the agent moves from one state to another as a result of its actions. The transition model helps the agent predict the outcomes of its actions. By understanding how its actions affect its state, the agent can make more informed decisions to maximize utility.

Learning agents

Learning agents continuously improve itself based on the experiences it gathers. Feedback forms an integral part of their learning curve which they store in their knowledge base. Learning enhances the agent's ability to operate in unfamiliar environments. On top of that, it uses a problem generator to design new tasks to train itself from collected data and past results. An interesting implementation of learning agents is personalized recommendation generators for eCommerce sites which over the time improves accuracy in product recommendation by learning from user purchase history and preferences.

Applications of learning agent

Learning agents are being utilized in a wide range of industries to transform operations and improve effectiveness. Here is an analysis of common applications in various sectors:

Autonomous Robots: Learning agents help robots become more adept at activities like navigation, manipulation, and human contact by allowing them to adjust to changing surroundings and gain experience.

Personalized Recommender Systems: By evaluating user behavior and preferences, learning agents drive recommendation engines in social networking, streaming services, and e-commerce platforms.

Financial Trading: To maximize trading tactics in financial markets, learning agents can evaluate market data, spot trends, and forecast future events.

Healthcare: To help medical practitioners make decisions, learning agents are employed in drug development, individualized treatment planning, medical diagnostics, and patient health data monitoring.

Game-Playing AI: From chess to video games, learning agents engage in strategic gameplay, honing their skills through self-play, human interaction, and iterative refinement.

Multi-Agent System

Multi-agent systems (MAS) are systems composed of multiple interacting autonomous agents. Each agent in a multi-agent system has its own goals, capabilities, knowledge, and possibly different perspectives. These agents can interact with each other directly or indirectly to achieve individual or collective goal

Characteristics of Multi-agent systems

Autonomous Agents: Each agent acts on its own based on its goals and knowledge.

Interactions: Agents communicate, cooperate, or compete to achieve individual or shared objectives.

Distributed Problem Solving: Agents work together to solve complex problems more efficiently than they could alone.

Decentralization: No central control; agents make decisions independently, leading to emergent behaviors.

Applications: Used in robotics, traffic management, healthcare, and more, where distributed decision-making is essential.

Google’s Perspective of AI Agents

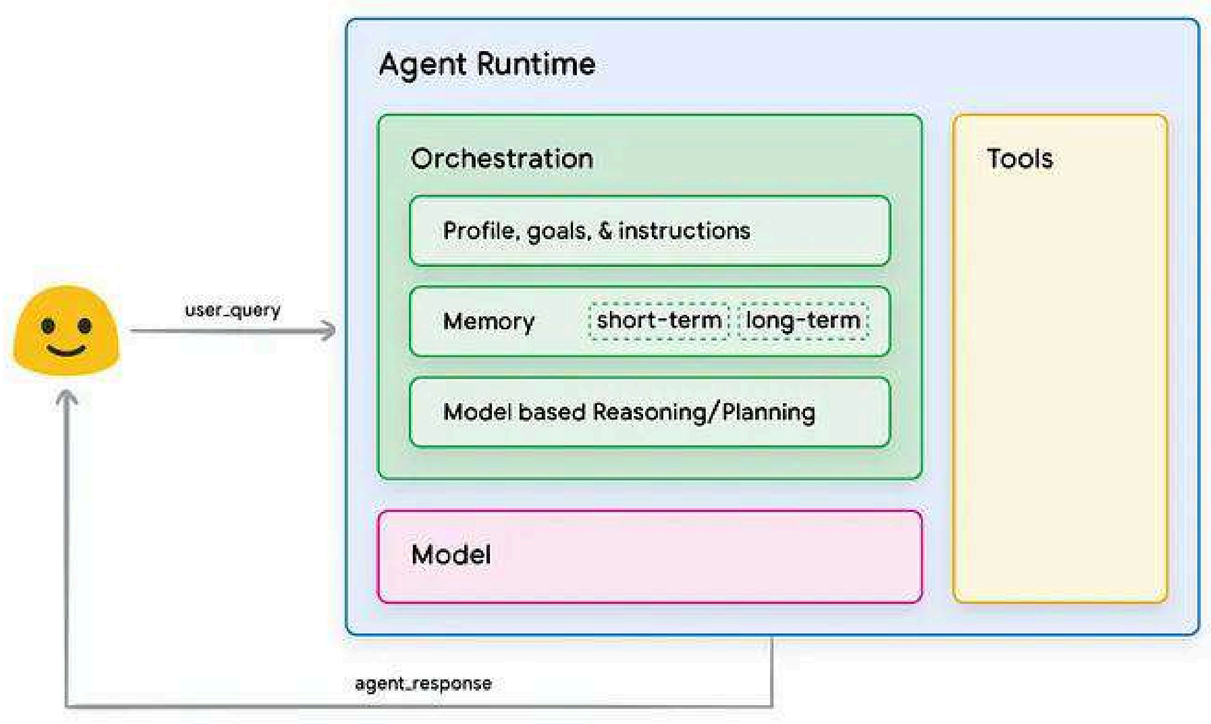
In September 2024, authors Julia Wiesinger, Patrick Marlow, and Vladimir Vuskovic published the groundbreaking white paper "Agents." This section provides a brief analysis and summary, highlighting its key insights.

Difference between AI models and AI Agents

AI Models in the context of AI Agents are Language Models (large/small) that are foundational components of an AI Agent. Models are incapable of interacting with the external world of their own. An AI Agent is an extension of the concept of this model that bridges a model to the external world with the help of tools to interact with external data and services to function autonomously combining reasoning, logic, and access to external information.

	AI Model	AI Agent
Knowledge Scope	Limited to training data.	Extends knowledge through external systems using tools.
Context Handling	Single inference, lacks session history management.	Maintains session history for multi-turn interactions.
Tool Integration	No native tool implementation.	Natively integrated tools for real-world interactions.
Reasoning Capability	No native logic layer, relies on user-defined prompts.	Built-in cognitive architecture with reasoning frameworks.

The tools of varying complexity usually align with common web API methods like GET, POST, PATCH, and DELETE to significantly expand the capabilities of foundational models. An orchestration layer governs how the agent takes in information, performs some internal reasoning, and uses that reasoning to inform its next action or decision. In general, this loop will continue until an agent has reached its goal or a stopping point.



General Architecture of an AI Agent (Source: “Agents” by Julia Wiesinger, Patrick Marlow and Vladimir Vuskovic)

Common frameworks for generative AI Agents

Agents use cognitive architecture to iteratively process information, and refine their decisions and actions based on previous output. The cognitive architecture is governed by the orchestration layer at its core. Orchestration layer leverages prompt engineering frameworks for reasoning and planning to ensure better interaction of the agent with the environment. Following is a list of the popular framework and reasoning technique in the contemporary AI landscape:

ReAct

As elaborated earlier in this white paper, a prompt engineering framework that combines reasoning and acting in an iterative process, enabling agents to use external tools or data sources dynamically while reasoning through tasks.

Chain of Thought (CoT)

A prompt engineering framework where the agent breaks down a complex problem into simple intermediate steps of reasoning. Variations of CoT techniques exist like self-consistency, active-prompt, and multimodal CoT.

Tree-of-thoughts (ToT)

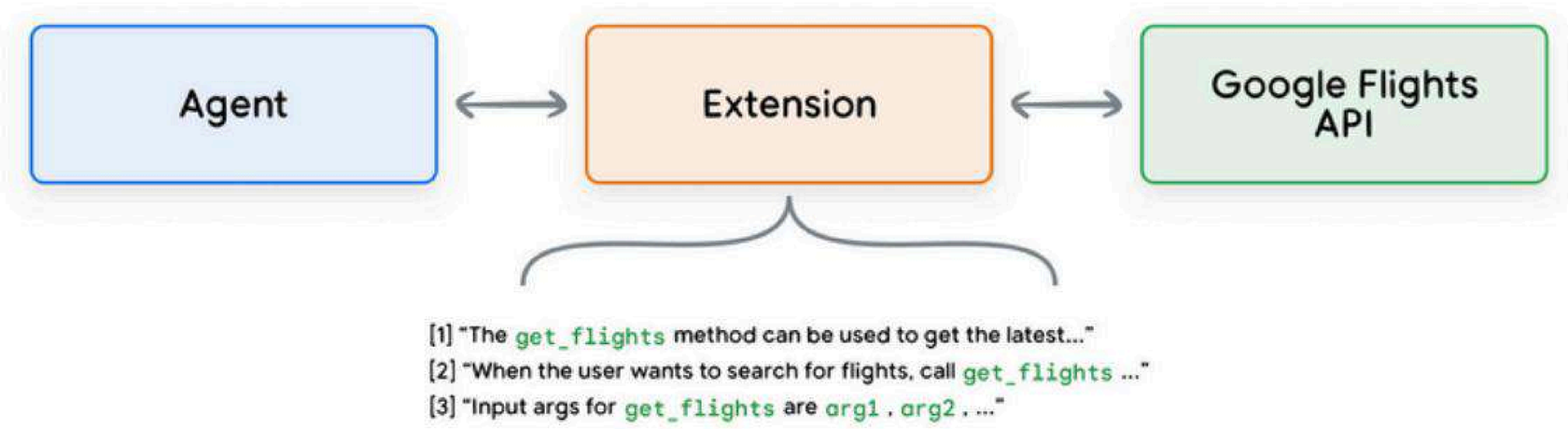
This framework extends on the CoT approach allowing models to explore multiple thought chains like a decision tree, enabling the agent to evaluate and choose optimal solutions. This framework is well suited for strategic or lookahead tasks

Primary types of tools

As stated earlier, tools are what connect a language model to the external world so that the agent can gather information and act in real-time. Following are the three primary types of tools that Google models can interact with:

Extensions

An Extensions is a bridge between an API and the agent that teaches the agent how to use the API endpoints and what arguments or parameters are needed to successfully call the API endpoint.



Example of an Extension connecting an Agent for flight booking to Google Flights API

Functions

These are reusable self-contained modules of code for performing a specific task. A function is executed on the client-side and the logic and execution of calling the actual API endpoint is offloaded away from the agent and back to the client-side. A Function varies very slightly from Extension where the later makes a direct API call. A function frees up the model from the complexity of API calling providing granular control to the developer or where API calls need to occur outside the Agent architecture for security constraints.

Data Stores

Data Stores enables developers to upload the latest data to the model workflow in the form of PDF or spreadsheet ensuring the model has access to up-to-date, dynamic information. The uploaded documents are converted into vector database embeddings ready for consumption by the agent to extract information and supplement its next action or response. This is particularly useful in Retrieval Augmented Generation (RAG) applications, where agents can search and retrieve real-time content from sources like websites or structured documents.

Targeted learning approach for enhanced model performance

Models require to develop specialized skills to use tools in production and learn in real-time to choose the right tools. Following is some of the learning approaches:

In-context Learning

This method provides the model with a prompt, tools, and few-shot examples encouraging it to learn ‘on the fly’ when and how to use a tool. The ReAct framework is a classic example of this learning approach.

Retrieval-based In-context Learning

Here the model gets dynamically populated with the relevant information, tools, and examples externally. RAG applications leveraging data stores or models connected with Extensions with example stores follow this approach.

Fine-tuning Based Learning

This method trains a model beforehand using large datasets helping it to understand when and how to apply certain tools prior to receiving any user queries.

Challenges of AI Agent

Technical complexity

Building AI Agents requires expertise in machine learning and deep learning technologies. Training and deploying AI Agents requires substantial computational resources. Multi-agent systems built on the same foundation models may experience shared pitfalls triggering a system-wide failure of all agents. Agents that are unable to create a comprehensive plan or reflect on their findings may find themselves repeatedly calling the same tools, invoking infinite feedback loops.

Data privacy

Training and operating AI Agents involves acquiring, storing, and moving massive volumes of data. This might lead to privacy and compliance concerns. Failure of multi-agent frameworks can leave all the agents vulnerable to external attack.

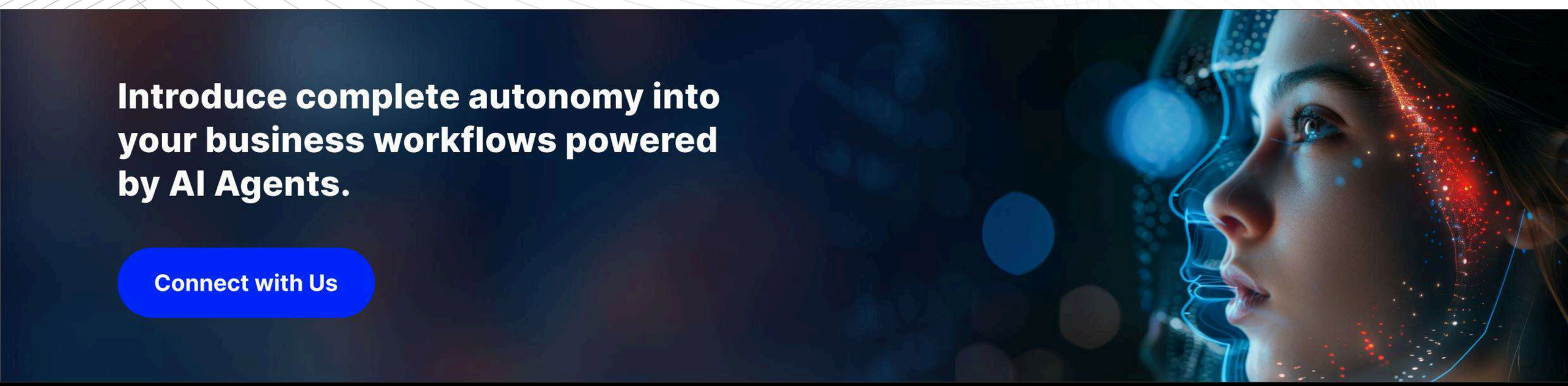
Ethical challenges

In certain circumstances, deep learning models may produce unfair, biased, or inaccurate results. Applying safeguards, such as human reviews, ensures customers receive helpful and fair responses from the agents deployed.

About Us

Gleecus Techlabs Inc. is one of the fastest growing IT innovation partners for startups, SMBs, and enterprises that help clients envision, build, and run more innovative and efficient businesses. We envision your business use cases for AI Agents and assist in achieving complete automation of workflows strategically bringing together 3 major concepts, i.e. machine learning (ML), natural language processing (NLP), and data engineering.

Our team specializes in developing AI Agents tailored to your unique needs. With years of expertise in AI, ML, NLP, GenAI, etc., our team can guide you through every step of the AI Agent development process, from ideation to deployment.



About Gleecus TechLabs Inc.

Gleecus TechLabs Inc. is an ISO 9001:2015 and ISO/IEC 20000-1:2018 certified Forward Thinking Digital Innovation partner creating impactful business outcomes with Engineering & Experience. With deep focus on Cloud, Data, Product Engineering, AI and Talent we help organizations become Digital Natives.

Gleecus TECHLABS
Forward Thinking Digital Innovation

Email: hello@gleecus.com

Phone: +1 347 947 2022

www.gleecus.com