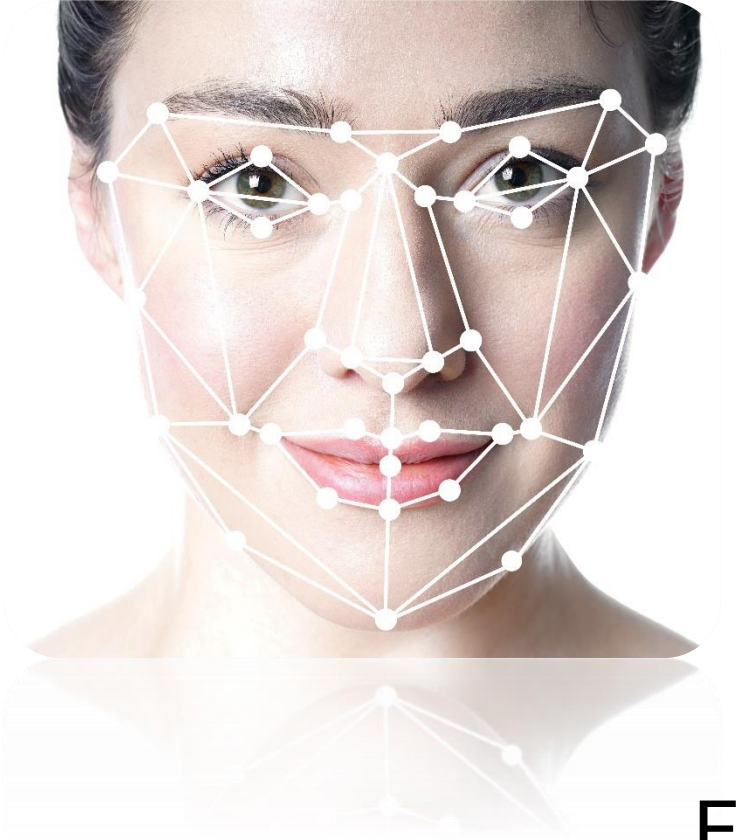
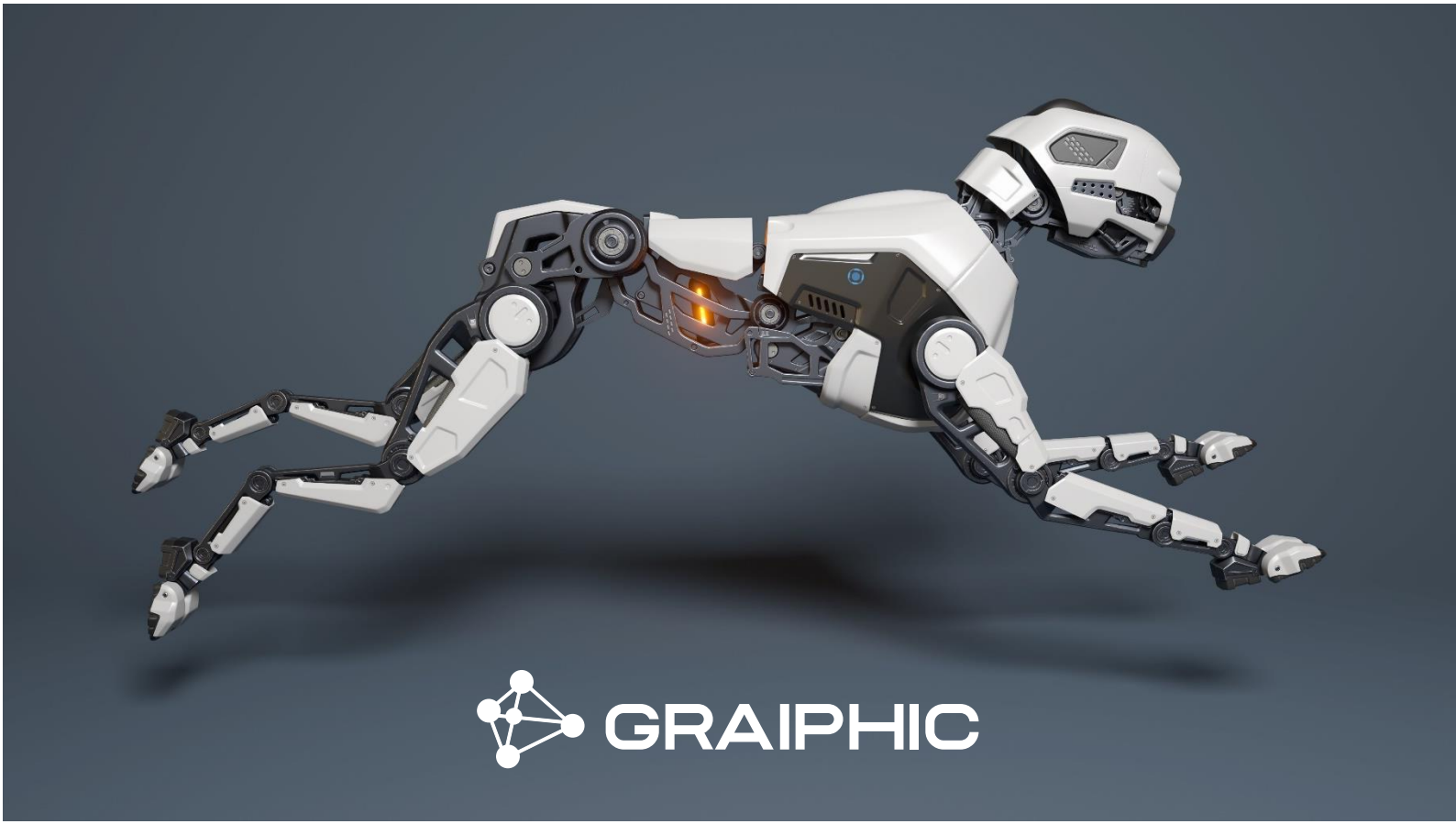


From Fragmentation to Orchestration: The GO GenAI Blueprint

Whitepaper 1.0



GRAIPHIC

Table of Contents

Versioning

This document is subject to version control to ensure full traceability of changes. Each update is recorded with its author, date, and a short description of the modifications.

Version	Date	Author	Organization	Change Description
1.0	2025/11/31	Youssef Menjour	Graiphic	First publication of the GO GenAI Whitepaper

Executive Summary

Generative AI is moving from experimentation to real deployment. Models are becoming smaller, faster, and more energy-efficient, hardware accelerators continue to improve, and the need for confidentiality and sovereignty is rising across all sectors. This evolution shifts the focus from large, centralized LLMs to agile SLMs that can run locally, securely, and with predictable performance. As industry, research, and academia rapidly expand their use of intelligent agents, the challenge is no longer model capability but the complexity of integrating, optimizing, and deploying these systems on diverse hardware.

Today's AI ecosystem is fragmented. Training, fine-tuning, serving, optimization, orchestration, and hardware execution all depend on different tools that rarely align. This fragmentation slows development, complicates reproducibility, and makes deployment difficult to scale or govern. Engineers and researchers need a unified way to design, functionalize, and deploy GenAI systems without navigating the maze of incompatible frameworks and scripting layers.

GO-GenAI answers this need by transforming ONNX into a universal, executable graph runtime where every component; models, tokenizers, logic, data streams, and hardware execution becomes a connected node within a single workflow. Graph Orchestration turns AI from isolated assets into systems that are structured, observable, and reproducible. One graph, one artifact, one deterministic execution path across GPUs, CPUs, NPUs, FPGAs, and sovereign edge environments.

GO-GenAI is introduced as a **new core component of the SOTA ecosystem**, fully integrated inside the LabVIEW-native environment that Graiphic provides. It is not added on top of SOTA, but built directly within it, becoming the part of the platform dedicated to Generative AI. Through SOTA's visual orchestration, engineers design, optimize, and deploy agents without Python, without scattered tools, and without glue code. The same interface used for instrumentation, automation, and control now becomes the unified environment for building complete GenAI systems.

This integrated architecture produces measurable outcomes. Provider-aware scheduling and quantization reduce latency and energy per token. Operator fusion and optimized memory handling deliver stable performance on any hardware. The same graph used for prototyping can be deployed unchanged into production. Compliance becomes native through deterministic seeds, shape locks, provenance tracking, and audit-ready execution logs. AI Act requirements are addressed by construction, not by retrofitting.

Across industries, GO-GenAI enables intelligent agents to become operational and reliable components. Manufacturing and robotics benefit from real-time reasoning integrated with control systems. Research labs gain reproducible experiments that can be shared as executable graphs. Academic programs adopt a transparent environment that clarifies AI mechanics. Healthcare, defense, and sovereign infrastructures deploy AI locally with full control over models, data, and logic.

By combining graph orchestration, hardware universality, visual system design, open-model compatibility, and guaranteed reproducibility, GO-GenAI sets a new standard for

Generative AI deployment. It turns intelligence into a unified, sovereign, and accessible component of real systems, fully integrated inside SOTA and ready for the next decade of industrial and scientific innovation.

Problem & Need

Generative AI is evolving quickly. Models are becoming smaller and more efficient, hardware accelerators continue to improve, and requirements for confidentiality and sovereignty are increasing across all domains. This shift places SLMs at the center of most industrial and academic workloads, making them the preferred choice for local, secure, and energy-efficient deployment. Yet despite this progress, the development of GenAI systems remains constrained by a fundamental issue: fragmentation.

Today, each stage of the AI lifecycle relies on a different toolset. Training requires one framework, fine-tuning another, serving and optimization rely on separate runtimes, orchestration depends on external libraries, and hardware execution is tied to vendor-specific SDKs. This creates a fragmented pipeline where every step adds complexity, multiplies dependencies, and reduces reproducibility. For engineers, researchers, and academic users, the challenge is no longer to build a model but to make it operate reliably, efficiently, and securely across heterogeneous environments.

Graiphic was created to eliminate these barriers. With SOTA, we unified deep learning, acceleration, and system design inside a single LabVIEW-native environment. GO-GenAI continues this mission at the Generative AI level. It is not an external extension or an additional layer, but a new core component fully integrated within the SOTA ecosystem. This integration transforms SOTA into a complete platform where AI systems can be designed, optimized, orchestrated, and deployed without relying on Python or scattered frameworks.

GO-GenAI brings every stage of the AI lifecycle into a continuous, reproducible workflow. Models, tokenizers, logic blocks, streams, and hardware execution paths become nodes of a single graph built entirely on ONNX and orchestrated visually inside SOTA. A model can be imported, adapted, retrained, functionalized, and deployed without leaving the same environment. The entire system becomes transparent, deterministic, and easy to reason about, turning Generative AI into an operational component rather than a collection of isolated scripts and tools.

The growing adoption of SLMs reinforces this need for unification. As architecture becomes more capable and more compact, the bottleneck moves from the model itself to the tooling around it. Users need a platform that is efficient, accessible, and able to deploy advanced agents on any hardware with consistent behavior and full sovereignty. GO-GenAI addresses this need by providing a universal graph runtime that merges computation, logic, compliance, and hardware scheduling into a single artifact.

The philosophy is clear: simplification through unification. GO-GenAI eliminates the technical fragmentation that slows innovation by bringing orchestration, execution, and deployment into one coherent system. GPUs, CPUs, NPUs, and FPGAs become

interchangeable targets inside the same graph. Development becomes intuitive, deployment becomes immediate, and optimization becomes measurable.

By integrating GO-GenAI directly into the SOTA ecosystem, Graiphic provides a complete and sovereign foundation for the next generation of intelligent systems. Users can work from concept to deployment without switching tools, rewriting code, or compromising on performance or compliance. GO-GenAI makes advanced Generative AI practical, transparent, and operational for industry, research, and academia.

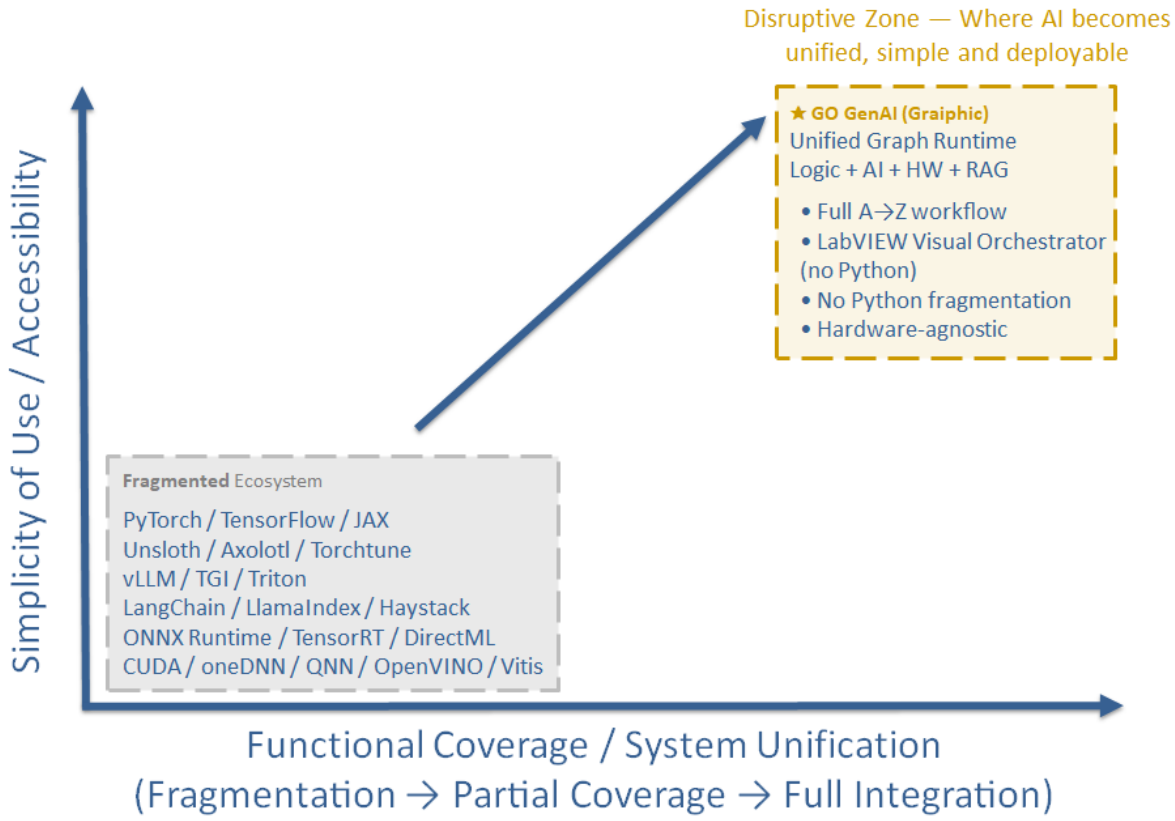


Figure 1. The current AI ecosystem is fragmented across multiple layers. Training, fine-tuning, inference, serving and hardware execution operate in isolated tools that do not form a unified system. GO GenAI is designed to introduce a single orchestrated runtime within SOTA, where models, logic and hardware become part of one continuous and reproducible workflow

State of the Art (SoTA) Where GO GenAI fits

Generative AI has reached a stage where open-source tools cover nearly every phase of the lifecycle. Training, fine-tuning, inference, orchestration, and deployment all benefit from an ecosystem that grows richer every month. This progress has accelerated innovation, yet it has also created a landscape marked by fragmentation. Each tool solves a specific part of the workflow, but none provides a true system-level foundation capable of turning models into complete and deployable intelligent agents.

Training frameworks such as PyTorch, TensorFlow and JAX remain central to research and model development. They offer flexibility and performance for experimentation, but they stop at the moment deployment begins. Once the training phase is complete, engineers must rebuild the entire pipeline using different runtimes, compilers, quantization tools and hardware-specific interfaces. This transition breaks continuity and introduces unnecessary complexity.

Distributed training toolkits push model capacity to impressive levels. Fine-tuning frameworks such as Unsloth, Axolotl, Torchtune or PEFT techniques allow efficient adaptation of powerful models including Qwen, Kimi, Mistral, Phi or DeepSeek. These tools are extremely effective in isolation, yet each operates in its own environment. Fine-tuning is performed in one tool, optimization in another, deployment in a third, and orchestration in a fourth. Every step multiplies dependencies and increases the cognitive load for users.

Inference and runtime engines have made remarkable progress. ONNX Runtime, TensorRT, DirectML, oneDNN, Vitis AI and llama.cpp deliver impressive execution performance on a wide range of hardware. They accelerate models such as Qwen, Phi, Llama, Kimi or Mistral on GPUs, CPUs, NPUs and edge devices. However, each engine focuses on single-model execution. They do not provide a unified view of tokenizers, memory, logic, preprocessing or multi-model routing.

Serving solutions such as vLLM, TGI, BentoML and Triton focus on throughput and scalable API deployment. Orchestration libraries such as LangChain, LlamaIndex, Haystack and Autogen bring higher-level logic by connecting models with tools and data sources. These solutions enable agent-like behavior, but they remain disconnected from the underlying runtime. As a result, they do not offer determinism, portability or reproducibility across hardware environments.

The overall ecosystem is rich in high-quality components. Each layer excels on its own, yet the layers do not naturally align. The result is a powerful but disjointed environment where training, fine-tuning, optimization, inference and orchestration live in separate worlds.

This is the space in which Graiphic introduces GO GenAI.
GO GenAI brings a new system layer built upon the ONNX ecosystem. Instead of treating fine-tuning, inference, logic, memory and hardware execution as isolated operations, GO GenAI connects them inside a single executable graph. Models, tokenizers, logic components, data streams and hardware execution paths all become interoperable nodes of one unified representation. The graph becomes the system, and the system becomes portable, reproducible and fully governable.

GO GenAI is introduced as a new core component of the SOTA ecosystem. It is integrated directly inside SOTA's LabVIEW-native environment and does not rely on Python or

external frameworks. With GO GenAI, SOTA evolves into a complete platform for Generative AI where users design, optimize, orchestrate and deploy intelligent agents visually and deterministically. Every component remains inside the same workflow and all logic becomes observable, measurable and ready for industrial deployment.

Unsloth accelerates fine-tuning. Llama.cpp enables efficient local inference. ONNX Runtime ensures hardware neutrality. LangChain and LlamaIndex bring high-level orchestration. GO GenAI unifies these advances inside one coherent system. One artifact describes the full intelligence workflow. One graph operates across CUDA, TensorRT, DirectML, OpenVINO, oneDNN, CPU, NPU and FPGA environments with consistent behavior and reproducibility.

GO GenAI is not an additional framework. It is the missing system layer that transforms isolated tools into a complete ecosystem. It provides a universal foundation where open-source models such as Qwen, Kimi, DeepSeek, Phi, Llama and Gemma can be trained, adapted, functionalized and deployed with total continuity across hardware and environments.

Graiphic's ambition is to bring the ecosystem from the age of fragmented tools to the age of unified intelligence. GO GenAI is the realization of that ambition inside SOTA, offering a simple, visual and universal way to build real Generative AI systems.

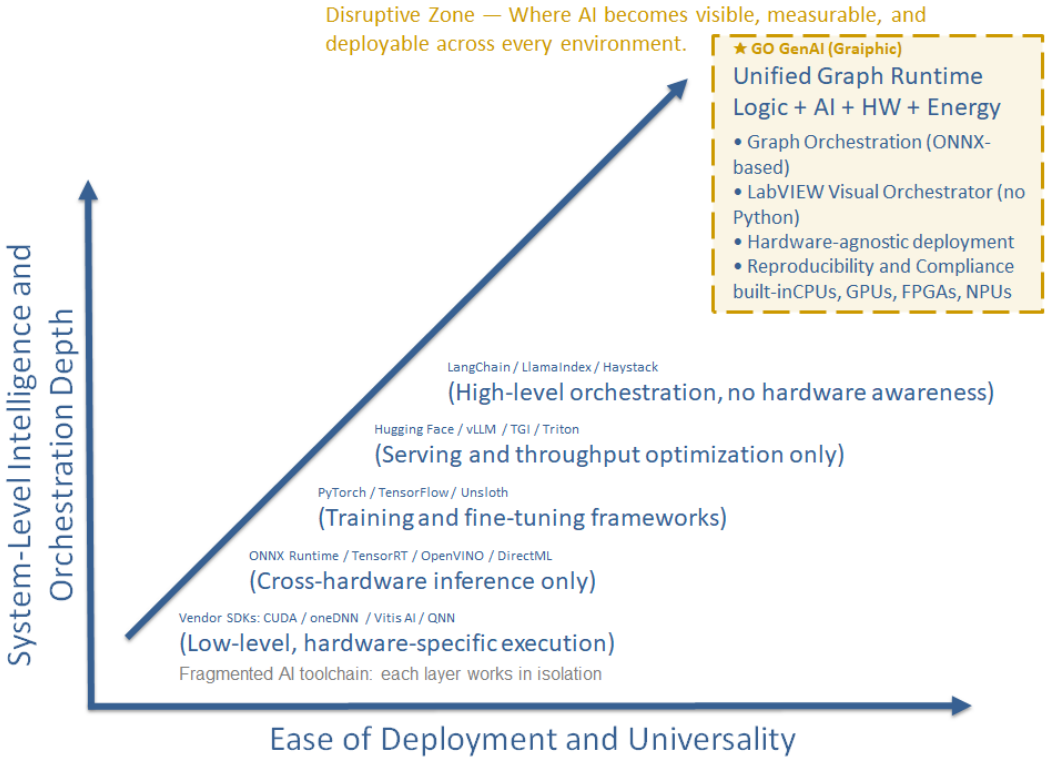


Figure 2. Open-source tools cover different aspects of the AI lifecycle, from low-level hardware execution to high-level orchestration. Each layer works independently and lacks system-level integration. GO GenAI aims to reach the upper right zone where intelligence, orchestration depth and deployment universality converge inside a single graph-based runtime integrated in SOTA.

Solution Overview GO GenAI

GO GenAI is conceived as the next major evolution of the SOTA ecosystem. It represents a new component that SOTA will integrate to extend its capabilities into the field of Generative AI. The goal of the project is to design a universal, graph-oriented runtime able to connect every element of an intelligent system inside a single, executable flow.

The central idea behind GO GenAI is Graph Orchestration. The project aims to transform the ONNX format from a static model container into a dynamic system representation where models, tokenizers, preprocessing and postprocessing steps, control logic and hardware execution paths are expressed as nodes in one coherent graph. This graph would be editable, connectable and optimizable directly within the LabVIEW-native environment of SOTA or through compiled C and C++ interfaces.

This approach follows Graiphic's philosophy of clarity and determinism. The project is designed to avoid Python and scripting layers, replacing them with visual orchestration directly inside the LabVIEW IDE. The ambition is to make AI pipelines transparent and measurable, where each part of the system is visible, inspectable and fully under control rather than hidden inside multiple scripts and frameworks.

GO GenAI is intended to enable a continuous workflow from concept to deployment inside a single environment. Researchers would be able to import or adapt models, engineers would be able to integrate them into control, automation or vision systems, and developers would be able to deploy them across GPUs, CPUs, NPUs or FPGAs without leaving SOTA. The same graph used during development could be executed without modification on workstations, industrial servers, embedded controllers or sovereign edge platforms.

From a technical perspective, GO GenAI plans to rely on ONNX Runtime and its broad selection of execution providers including TensorRT, DirectML, OpenVINO, oneDNN and Vitis AI. The orchestration engine would distribute subgraphs according to hardware capabilities, optimizing each segment for latency, energy consumption and memory efficiency. Automated quantization, batching and operator fusion at the C++ layer are part of the project's performance objectives.

Once integrated into SOTA, GO GenAI would provide an intuitive visual environment where engineers import, assemble, measure and deploy AI systems using the same principles they apply to LabVIEW instrumentation. Every graph block would represent a clear system component such as a model, a tokenizer, a logic unit or a hardware execution node. This unified view is intended to make advanced AI pipelines understandable, maintainable and reproducible.

GO GenAI is therefore envisioned as a new generation AI runtime that combines performance, simplicity and compliance inside a single visual paradigm. By integrating it

into SOTA, Graiphic aims to give research laboratories, industrial groups and academic institutions a platform where intelligent agents can be designed faster, deployed more reliably and governed with full sovereignty. The project formalizes Graiphic's long-term vision: a unified, open and sovereign AI ecosystem where intelligence becomes an integrated component of every system.

Technical Deep Dive

GO GenAI is conceived as a new system layer dedicated to the next generation of open-source models. Its purpose is to make high-performing architectures such as Kimi K2, Qwen 2.5, DeepSeek V3 and gpt-oss directly usable, reproducible and deployable across a wide range of hardware and industrial environments. These models demonstrate that open innovation has matured. They deliver reasoning and contextual understanding comparable to proprietary systems while remaining transparent and accessible. Yet they still depend on separate toolchains, serving stacks and optimization pipelines. GO GenAI aims to remove this fragmentation by designing a single unified runtime capable of executing, orchestrating and functionalizing open models inside the SOTA ecosystem.

At its core, the project extends ONNX into a living graph runtime. Every component of an intelligent agent such as the model, tokenizer, logic, preprocessing, postprocessing and hardware execution becomes a node inside one executable graph. The Graph Orchestration Engine coordinates these nodes deterministically, handling both numerical computation and system-level decision logic. Instead of scattered scripts and frameworks, the entire agent is expressed as a clear and inspectable structure.

The architecture distinguishes two complementary layers.
Data Flow handles numerical execution. It manages tensors, operators and memory transfers across heterogeneous backends including CUDA, TensorRT, DirectML, oneDNN, OpenVINO, CPU and FPGA.

Control Flow governs orchestration and sequencing. It defines how subgraphs execute and interact, enabling dynamic routing between models such as using Qwen for reasoning, Kimi for dialogue or DeepSeek for extended-context inference. This separation preserves static optimization for mathematical workloads while enabling runtime flexibility for agent behavior. Engineers can modify workflows or insert new models without re-exporting the entire graph.

GO GenAI is designed to incorporate provider-aware scheduling. Each node in the graph can be mapped to the most efficient execution provider according to latency, energy constraints and memory availability. This design ensures reproducible execution across various hardware profiles and prevents unpredictable behavior during deployment.

The runtime introduces several graph-level optimization strategies.

Operator fusion groups attention, MLP and normalization steps across subgraphs to minimize overhead.
Shared buffers reduce memory movement between successive model components. Quantization paths to INT8, FP8 or 4-bit weights target efficient inference for both large and compact models.

A KV-cache manager orchestrates host-device paging to support long-context reasoning and multi-turn interactions without exhausting GPU memory.

At the developer level, GO GenAI is designed to provide a fully compiled C and C++ API. This low-level interface integrates seamlessly with Graiphic's SOTA environment and with embedded or industrial systems where determinism and reproducibility are essential. The project introduces a simple graph-building workflow in LabVIEW, letting engineers assemble models, tokenizers, logic and hardware paths visually. All components generate a single ONNX-based artifact that can be executed identically across devices.

A typical workflow illustrates this process.

1. A model such as Qwen 2.5 is adapted or fine-tuned with existing open-source tools.
2. The resulting checkpoint is exported to ONNX.
3. Inside SOTA, the engineer composes the full agent by connecting tokenizers, preprocessing steps, model nodes and orchestration logic through the LabVIEW visual environment.
4. The unified graph is deployed on GPU, CPU or FPGA without modification.
5. Latency, throughput and energy metrics are observed directly through SOTA's instrumentation tools.

This unified approach transforms model integration into a deterministic and measurable process. Open models such as Kimi, Qwen, DeepSeek and gpt-oss become interoperable components inside the same runtime. Engineers design once, deploy anywhere and retain full visibility over execution behavior.

GO GenAI therefore establishes the missing foundation that turns individual models into complete, orchestrated systems. By integrating it natively into the SOTA ecosystem, the project aims to make open-source intelligence accessible, auditable, reproducible and fully hardware agnostic.

Call to collaborate

We invite hardware vendors, system integrators, safety-critical partners, and early adopters to co-define coverage, benchmarks and certification pathways. GO-GenAI's open-core vision aims to deliver a durable and portable foundation for Generative AI systems: one graph, many targets, predictable results. This initiative is also open to strategic funding and investment discussions with organizations that wish to support and accelerate the development of a sovereign, unified AI infrastructure.

Graiphic has built the first end-to-end ecosystem where AI, logic, and hardware orchestration live inside a single ONNX graph.

We are now opening a Call for Funding to accelerate the development and roadmap of Graiphic's GO GenAI initiative.

Why Invest?

- **Strategic Advantage:** Gain early access to the first unified runtime that connects models, logic and hardware execution inside a single graph, deployable across CPUs, GPUs, FPGAs, NPUs and edge devices.
- **Portability and Standards:** Ensure your hardware, SDKs and platforms are natively supported in a framework designed to become the open standard for orchestrated Generative AI.
- **Efficiency and Energy Insight:** Benefit from integrated performance and energy instrumentation directly at graph level, enabling precise optimization and cost-efficient deployment across devices.
- **Safety and Trust:** Contribute to shaping the next generation of reproducible, traceable and audit-ready AI workflows, supporting certification needs in aerospace, defense, automotive, healthcare and critical infrastructure.
- **Market Reach:** From embedded boards to cloud servers, GO GenAI is designed for any environment. Your technology can be part of a universal orchestration ecosystem adopted by industry, research and academia.

What We Offer

- Co-development opportunities with Graiphic's engineering and R&D teams.
- Early integration of your hardware, SDKs and acceleration backends into GO GenAI.
- Joint visibility in international standardization efforts (ONNX, Horizon Europe, DARPA-aligned initiatives).
- Shared benchmarking and open-source dissemination to establish your platforms as reference solutions for orchestrated Generative AI.
-

How to Engage
Graiphic is actively seeking:

- Equity investors who want to support and accelerate the development of a sovereign AI orchestration ecosystem.
- Industrial sponsors willing to co-fund R&D programs, prototyping cycles and test benches.
- Strategic partners (hardware vendors, system integrators, technology providers, major OEMs) wishing to position their platforms at the core of the future ONNX-based Generative AI ecosystem.

Join us in shaping the universal cockpit for AI.
Contact: funding@graiphic.io | www.graiphic.io

Annexes

Support Letters

GO GenAI is already supported by more than ten organisations across Europe, including industrial groups, academic hospitals, research laboratories, and engineering companies. These partners spanning France, Belgium, Italy, Germany and the UK have each provided formal letters of endorsement demonstrating strong interest in the deployment, validation, and adoption of the GO GenAI approach. All support letters are included in the annex.

UNIVERSITÀ
DEGLI STUDI
DI MILANO

Università degli Studi di Milano
Via Festa del Perdono, 7
20122 Milano, Italy
Phone: +39 02 5032 5032

September 13, 2025

Subject: Letter of Support for Graiphic's GO GenAI (ONNX-GenAI) Project

To whom it may concern,

The **Università degli Studi di Milano** is a leading Italian research university with a strong tradition in clinical research and translational medicine. Our community values open, auditable, and efficient technologies that can be adopted safely within European regulatory frameworks.

We are pleased to support **GO GenAI** by **Graiphic**. GO GenAI builds upon the ONNX standard and **ONNX Runtime** to deliver an open, **MIT-licensed**, hardware-agnostic orchestration layer for generative AI. Its **graph orchestration (GO)** turns ONNX into a programmable execution environment where models, operators and control flows can be composed into full, deterministic and energy-efficient pipelines—appropriate for hospital IT/OT integration and compliance with the **EU AI Act, GDPR and MDR**.

For intensive care and imaging-supported diagnostics, this approach could enable portable, reproducible and efficient deployments from servers to embedded NPUs, accelerating evaluation and clinical readiness while preserving data sovereignty.

Pending standard approvals, we are open to scientific collaboration on benchmarking, clinical workflow integration and dissemination activities. This letter entails no financial or exclusivity commitment.

Sincerely,

Dr. Davide Chiumello
Università degli Studi di Milano
Email: davide.chiumello@unimi.it

The AI Data Robotics Association

Boulevard Auguste Reyers 80. 1030 Schaerbeek. Belgium
Phone +32 465 85 0760
Web www.adr-association.eu
Email secretary-general@adr-association.eu
TVA BE0768619684

Brussels, 15.09.2025

Letter of Support for GO GenAI

To Whom It May Concern,

On behalf of the **AI, Data and Robotics Association (ADRA)**, the European public-private partnership committed to advancing trustworthy, sovereign, and energy-efficient AI, we are pleased to express our strong support for **GO GenAI**, developed and led by Graiphic.

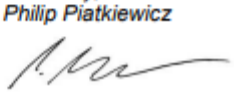
ADRA's mission is to **strengthen Europe's leadership in AI, Data, and Robotics** by fostering an ecosystem that combines **digital sovereignty, industrial competitiveness, and societal trust**. GO GenAI is fully aligned with these objectives:

- It advances **digital sovereignty** by building an open and sovereign generative AI runtime, anchored in the global ONNX standard.
- It enhances **energy efficiency** through optimized execution, enabling Europe to reduce the compute and carbon footprint of next-generation AI models.
- It ensures **trust and compliance by design**, directly supporting Europe's implementation of the **AI Act**.

By endorsing GO GenAI, ADRA recognizes Graiphic's contribution to **Europe's technological autonomy** and its unique role as one of the few EU-born companies actively shaping the future of **ONNX and ONNX Runtime**, projects that currently remain dominated by the US.

We strongly support GO GenAI and look forward to seeing the impact of Graiphic's work for **Europe's AI ecosystem, its citizens, and its industry**.

Sincerely,
Philip Piatkiewicz



Secretary-General
Signature

Ouassila Labbani Narsis
CIAD Laboratory - University of Bourgogne Europe
64 Rue de Sully, 21000 Dijon, France
+33 380 39 58 72

September 24, 2025

Subject: **Support for Graiphic and GO GenAI (ONNX-GenAI)**

To whom it may concern,

The CIAD Laboratory (UR 7533, University of Bourgogne) is pleased to express its strong support for Graiphic and its GO GenAI initiative, also known administratively as ONNX-GenAI.

Our research group has a long-standing expertise in **Informed Machine Learning (IML)**, **agent-based AI**, and **knowledge engineering**, with numerous scientific contributions in domains such as oncology, industrial processes, and autonomous systems. CIAD has developed methodologies that combine domain knowledge, symbolic representations, and advanced learning techniques to produce AI models that are **transparent, explainable, and aligned with human expertise**. In parallel, our work on **agentic AI** demonstrates how intelligent agents can integrate reasoning, planning, and interaction capabilities for complex, dynamic environments.

We see a natural synergy between these research directions and Graiphic's **GO GenAI** project, which extends the ONNX Runtime into a sovereign and energy-efficient orchestration layer for generative AI. By enabling graph-level orchestration of Small and Large Language Models (SLMs/LLMs), decision logic, and hardware control, Graiphic's technology provides a unique foundation for transparent, frugal, and compliant AI systems. This aligns perfectly with our mission to design AI that is not only high-performing but also **responsible, auditable, and trustworthy**.

CIAD is committed to supporting Graiphic's efforts and to exploring future collaborations around GO GenAI. We believe this initiative will accelerate the integration of **IML principles** into industrial-grade toolchains, foster the development of **explainable and domain-aware generative AI**, and strengthen European sovereignty in AI infrastructure.

We therefore endorse Graiphic's vision and confirm our interest in contributing scientific expertise and collaborative research to the GO GenAI ecosystem.

Sincerely,

Ouassila Labbani Narsis
Associate Professor, Project Coordinator
CIAD Laboratory, University of Bourgogne Europe
ouassila.narsis@u-bourgogne.fr

Guy's and St Thomas'
NHS Foundation Trust

Guy's & St Thomas' NHS Foundation Trust
Trust Office, 4th Floor, Gassiot House
St Thomas' Hospital, Westminster Bridge Road
London SE1 7EH, UK
Phone: +44 (0)20 7188 7188

September 13, 2025

Subject: Letter of Support for Graiphic's GO GenAI (ONNX-GenAI) Project

To whom it may concern,

Guy's & St Thomas' NHS Foundation Trust, together with King's College London, is an internationally recognized academic healthcare centre delivering world-class critical care and respiratory medicine. Our teams routinely evaluate innovations that can improve patient outcomes, clinical efficiency, and compliance within regulated environments.

We are pleased to express our support for **GO GenAI**, led by **Graiphic**. GO GenAI proposes a sovereign, open and hardware-agnostic runtime for generative AI built on **ONNX** and **ONNX Runtime**, enabling graph-level orchestration across CPUs/GPUs/NPUs/FPGAs with on-premises deployment, auditability and alignment with the **EU AI Act, GDPR and MDR**. The project's **GO (Graph Orchestration)** layer turns ONNX from a static model format into a programmable execution graph for complete AI workflows, not just single-model inference, with efficiency and determinism suitable for clinical settings.

In critical care and imaging-driven care pathways, we see potential for reproducible on-prem LLM/VLM workflows, robust pre/post-processing, and transparent decisioning, supporting faster time-to-report and energy-efficient inference without vendor lock-in.

Subject to standard ethical approvals and institutional processes, we are open to collaborating on scientific validation and clinical evaluation activities, and to contribute expertise to benchmarking, integration and dissemination, without any

financial or exclusive commitment implied by this letter.

Sincerely,

Prof. Luigi Camporota
Consultant in Intensive Care / Professor of Critical Care Medicine
Guy's & St Thomas' NHS Foundation Trust & King's College London
Email: luigi.camporota@kcl.ac.uk

LA FRANÇAISE REM
Crédit Mutuel Alliance Fédérale

128, boulevard Raspail – 75006 PARIS
Tél. : + 33(0)1 73 00 73 00

Mr. Guillaume Allard

La Française

128, boulevard Raspail 75006 Paris, France

Tel: +33 (0)1 40 39 20 00

September 29, 2025

Subject: **Support for Graiphic's GO GenAI**

To whom it may concern,

As an executive board member of **La Française Real Estate Managers**, a subsidiary of the Crédit Mutuel Alliance Fédérale Group — the 5th largest French asset manager — I express my support for the **GO GenAI project** proposed by Graiphic.

Today, artificial intelligence is emerging as the next technological revolution. Yet, in practice, most of the applications we have seen so far remain disappointing, particularly when measured in terms of efficiency, reliability, and long-term impact. In our own field of **building energy management**, our teams have observed that existing AI-based solutions often suffer from critical shortcomings: black-box approaches that cannot be explained, systems that require constant retraining, or models that fail to adapt when faced with new operating conditions. The result is too often unreliable, energy-inefficient, and ultimately costly solutions — far removed from the promises initially made.

LA FRANÇAISE REM
Crédit Mutuel Alliance Fédérale

128, boulevard Raspail - 75006 PARIS
Tél. : + 33(0)1 73 00 73 00

Graiphic recently approached me to discuss potential collaborations and presented its vision, including **GO GenAI**. In simple terms, GO GenAI is a new orchestration layer that makes advanced AI models portable, explainable, and adaptive across diverse hardware and software environments. This innovation goes far beyond incremental improvements: it creates the conditions for trustworthy and efficient AI, capable of being integrated into real-world infrastructures, where reliability and transparency are paramount.

Faithful to our group's longstanding commitment to **responsible innovation** — and especially to projects rooted in **French and European excellence** — I see in Graiphic's work the potential to bring tangible benefits to our industry and to society at large.

For these reasons, I am pleased to support Graiphic and its GO GenAI project. Once the technology has reached maturity, I sincerely hope that Graiphic will be able to offer La Française and its partners an **innovative, reliable, and high-performance energy management solution**, aligned with our environmental, social, and governance (ESG/ISR) objectives.

Yours faithfully,

Mr. Guillaume Allard

Executive board member, La Française Real Estate Managers

guillaume.allard@la-francaise.com

MQ Consult
Branichstrase 66
69198 Schriesheim
Phone: +49 172 6214723
Email: mquintel@gwdg.de

September 19, 2025

Subject: Letter of Support for Graiphic's GO GenAI (ONNX-GenAI) Project

To whom it may concern,

I was the former head of the Department of Anesthesiolgy and Intensive Care Medicine at the University of Göttingen Medical Center (**UMG)** and still keep a part-time research position at this institution. As the head of this department at the UMG, a leading academic hospital, I was always advancing research and clinical innovation in my specialty anesthesiology and intensive care valuing and developing technologies that combine performance, high validity and reproducibility and by that improve outcome and patient safety. Since my retirement I continued with my research activities focusing on acute lung failure and started my own little company that coordinates research activities, consults and, promotes self-developed tools that support medical decision making in daily clinical practice. For almost my whole academic life I'm working together with Peter Herrmann (Herrmann MSE), in the early 90ties we developed a diagnostic software tool to analyze thorax CT scans to characterize acute lung failure in depth (MALUNA) which we continuously refined.

I support **GO GenAI** by **Graiphic**, a sovereign, ONNX-native orchestration runtime that brings **deterministic, graph-level control** to generative AI, with **hardware-agnostic** deployment (CPUs, GPUs, NPUs, FPGAs) and full auditability under European regulations. The approach is consistent with Graiphic's prior delivery of operative AI tooling in clinical environments and aligns with our needs for on-premises, transparent and energy-aware execution.

Subject to standard approvals, I'm open to support the scientific validation and clinical evaluation activities, and to contribute with my expertise to benchmarking, integration and dissemination. This letter is non-binding and involves no financial or exclusivity commitment.

Sincerely,

Prof. Dr. Michael Quintel

herrmann
MSE
Medical Software Engineering

Dr.sc.hum. Peter Herrmann, Dipl.-Ing (FH)
Am Grebenberg 8b, 37176 Nörten-Hardenberg
Phone: +49 176 99766557

September 16, 2025

Subject: Letter of Support for Graiphic's GO GenAI (ONNX-GenAI) Project

To whom it may concern,

I have been working in intensive care research for about 30 years. At the University Hospital Göttingen (**UMG**), my work includes developing software for our clinical and experimental research projects. My accumulated knowledge of interpreting and analyzing computed tomography images of the lungs led to the founding of **herrmann-MSE**.

Since 2022, we have been successfully collaborating with **Graiphic** (France), including on automated **lung CT segmentation** in real-world experimental and clinical workflows in our intensive care department. The SOTA/ONNX toolchain consolidated AI development and image processing, achieving measurable accuracy improvements in iterative cycles. Building on this, we support **GO GenAI**, which extends ONNX with **graph orchestration** for end-to-end generative-AI pipelines, enabling **on-premises**, auditable, and energy-efficient execution across heterogeneous hardware via ONNX Runtime **Execution Providers**.

The **UMG** as Part of the Georg August University of Göttingen, it is a major university hospital with a strong track record in the fields of anesthesiology, intensive care medicine, and medical image analysis. **herrmann MSE** is a small, emerging company specializing in the development of medical CT analysis software. The development of innovative AI tools using the Graphical SOTA/ONNX ecosystem plays a key role.

We recognize the potential impact on clinical efficiency and reproducibility and are open, subject to usual approvals to contribute to scientific validation, benchmarking against state-of-the-art, and dissemination. This letter does not imply financial or exclusive commitments.

Sincerely,

Dr. Peter Herrmann
herrmann-MSE
Email: peter@herrmann-mse.de
Phone: +49 176 99766557

Göttingen University Hospital (UMG)
Department of Anesthesiology
pherrmann@med.uni-goettingen.de
+49 551 3963834

PMB – Head Office
Route des Michels (CD56), Lieu-dit « La Corneirelle »
13790 Peynier, France
Phone: +33 (0)4 42 53 13 13

September 15, 2025

Subject: Letter of Support for Graiphic's GO GenAI (ONNX-GenAI) Project

To Whom It May Concern,

PMB ALCEN designs and manufactures advanced particle-accelerator systems and high-technology components for medical and industrial applications. Operating from Peynier (Aix-Marseille area), PMB brings rigorous engineering, quality, and safety culture to complex, highly regulated markets.

We support **GO GenAI** by **Graiphic**. Building on the **ONNX** standard, GO GenAI provides a programmable, hardware-agnostic orchestration runtime for generative AI that can be deployed on-premises with traceability and reproducibility. For PMB's environment, we foresee value in engineering documentation assistants, procedures and test-bench automation, knowledge capture from legacy datasets, and operator guidance with multi-modal models, while preserving IP and data sovereignty.

Subject to our internal processes and applicable regulations, we are open to collaborating on pilots, benchmarking and knowledge-transfer activities. This letter is non-binding and creates no financial or exclusivity obligation.

Sincerely,
Nicolas Massé
Nuclear Medicine Business Unit Director (groupe Alcen)
nmasse@pmb-alcen.com

Santerne – Vinci Énergies
1 avenue Paul Héroult
13015 Marseille, France
Phone: +33 4 91 09 56 30

September 15, 2025

Subject: Letter of Support for Graiphic's GO GenAI (ONNX-GenAI) Project

To Whom It May Concern,

Santerne, a **VINCI Énergies** company, is a long-standing French brand in electrical engineering, climate engineering and multi-technical services, supporting energy-efficient buildings, industrial facilities and digital infrastructures. Our operational culture values safety, quality, and pragmatic innovation at scale.

We support **GO GenAI** by **Graiphic**. The project brings **ONNX-based, hardware-agnostic orchestration** for generative AI workflows that can run **on premises**, with full audit trails and strong alignment to European regulatory expectations. Deterministic graph execution and edge-friendly performance make GO GenAI relevant for building operations, facility management, and industrial maintenance.

Expected benefits include: LLM/VLM assistants for commissioning and maintenance, automated technical documentation, anomaly detection and triage across BMS/EMS data, and portable deployments from the datacenter to embedded edge devices.

Pending our usual governance steps, we are open to participate in PoCs and technical evaluations. This letter is non-binding and implies no financial or exclusivity commitment.

Sincerely,
Olivier Malara
Santerne – VINCI Énergies
olivier.malara@santerne-marseille.com

Santerne Marseille
1, Avenue Paul Héroult - 13015 MARSEILLE
Tél. 04.91.09.56.30 - Fax 04.91.09.56.31
SAS AU CAPITAL DE 700.000 € - RCS MARSEILLE B 439 467 804

SATELEC – Fayat Énergie Services
24, avenue du Général de Gaulle
91178 Viry-Châtillon Cedex, France
+33 (0)1 69 56 56 56

September 15, 2025

Subject: Letter of Support for Graiphic's GO GenAI (ONNX-GenAI) Project

To Whom It May Concern,

Satelec, a Fayat Énergie Services company, delivers multi-technical excellence in electrical engineering, renovation, and maintenance for buildings and infrastructures across France. Our teams support public and private customers in demanding environments where reliability, energy performance, and operational continuity are paramount.

We are pleased to express our support for **GO GenAI** by **Graiphic**. GO GenAI proposes an ONNX-native, hardware-agnostic orchestration runtime for generative AI, designed for on-premises deployment, data sovereignty, and auditability. Its graph-level control enables deterministic, efficient pipelines that integrate pre/post-processing and control logic, suitable for industrial and smart-building contexts.

Potential benefits for Satelec's ecosystem include: documentation assistants and code-generation for PLC/SCADA integration; multi-modal troubleshooting (text, image, sensor data); energy-aware optimization; and deployment from servers to edge NPUs/GPUs, without vendor lock-in.

Subject to our standard internal reviews, we are open to collaboration on pilots (PoCs), benchmarking, and dissemination activities. This letter is non-binding and entails no financial or exclusivity commitment.

Sincerely,
Mr. Romain-Gaël Richard
Pole Director
SATELEC – Fayat Énergie Services
rg.richard@satelec.fayat.com

University of Turin,
Department of Surgical Science
Via Verdi, 8
10124 Torino, Italy
Phone: +39 011 670 6111

September 13, 2025

Subject: Letter of Support for Graiphic's GO GenAI (ONNX-GenAI) Project

To whom it may concern,

The **University of Turin** is one of Italy's oldest and most respected universities, with active programmes in biomedical and translational research. We are committed to advancing open, auditable and efficient AI technologies that can be deployed responsibly in regulated environments.

We support **GO GenAI**, developed by **Graiphic**. By extending the **ONNX** standard with **graph orchestration**, and integrating tightly with **ONNX Runtime**, the project enables complete, programmable AI workflows, covering pre/post-processing, multi-model composition and control logic, running **on-premises** across heterogeneous hardware while facilitating **EU AI Act/GDPR/MDR** compliance. The stack is open-source (MIT) and designed for deterministic, energy-efficient execution.

Subject to usual institutional procedures, we are open to collaboration on scientific validation, benchmarking and dissemination.
No financial or exclusivity obligations are implied by this letter.

Sincerely,

Francesca Collino

Dr. Francesca Collino
University of Turin
Email: francesca.collino@unito.it

State of the Art in AI Agent Development: Needs, Solutions, and Relevance of GO GenAI

Introduction

Artificial intelligence has rapidly advanced with **large language models (LLMs)** like GPT-4 and ChatGPT demonstrating unprecedented capabilities. However, the AI revolution is no longer just about ever-bigger models. There is a growing recognition that **smaller, specialized models** can often be more efficient and practical for real-world use. In particular, concerns about data privacy, deployment cost, and domain-specific accuracy are driving a shift toward **small language models (SLMs)** – more compact models tailored to specific tasks or data[1][2]. Simultaneously, organizations across industry and academia are seeking ways to integrate these AI models into their workflows securely and effectively. This has highlighted the need for new tools and frameworks that make it easier to **build, deploy, and maintain AI "agents"** (autonomous AI-driven processes) with *speed*, *security*, and *accessibility*. In this context, solutions that simplify programming (e.g. through graphical interfaces), ensure data confidentiality (on-premise or private deployments), and streamline the end-to-end pipeline are becoming essential. The convergence of more efficient models, powerful GPU acceleration, and a demand for trustworthy AI has created fertile ground for innovation. **GO GenAI**, Graiphic's new platform, emerges as a response to these trends – aiming to unify and simplify AI agent development. Before detailing its significance, we first survey the state-of-the-art: current needs, existing solutions, market trajectory, and the challenges faced in deploying AI agents today.

From Large to Small Language Models (LLMs to SLMs)

For the past few years, progress in AI was often equated with ever-larger language models trained on massive datasets. These **LLMs** are incredibly versatile and capable of general-purpose reasoning, but they demand enormous computational resources and can be unwieldy to fine-tune or deploy. **Small language models (SLMs)**, by contrast, have far fewer parameters and are usually fine-tuned for narrower domains or tasks[3][4]. Key differences have been noted in recent literature and industry analyses:

- **Efficiency vs Versatility:** LLMs deliver broad capabilities across many domains, whereas SLMs excel in specialized fields (e.g. medicine, law, finance) where precision is paramount[1]. Because SLMs focus on specific data, they can be more **resource-efficient**, requiring less memory and computing power to run.
- **Resource and Data Constraints:** LLMs typically require large-scale cloud infrastructure and huge training datasets, raising concerns about cost and data privacy. SLMs, on the other hand, can often be deployed on-premises or on edge devices, keeping data local and secure while still performing well on targeted tasks[1]. This aligns with a broader enterprise need to avoid sending sensitive data to third-party cloud APIs.
- **Performance in Context:** While a giant LLM might outperform a small model in open-ended conversations or knowledge breadth, a well-crafted SLM can actually **outperform an LLM on domain-specific tasks** due to having seen more

relevant data for that task. Moreover, SLMs may carry lower risk of off-topic "hallucinations" or bias because they operate within a constrained knowledge scope[5].

Notably, the industry is recognizing that *"smaller is smarter"* in many scenarios. A **2025 Harvard Business Review** article sums up this shift: *"According to a recent study by NVIDIA researchers, small language models (SLMs), rather than their larger counterparts, could become the true backbone of the next generation of intelligent enterprises. The age of 'bigger is better' may be giving way to 'smaller is smarter.'"*[6]. Those NVIDIA researchers argue that SLMs are sufficiently powerful for most tasks in *agentic AI systems* and are inherently more suitable and economical to deploy at scale[7][8]. In practical terms, an SLM (say a few billion parameters) can often be fine-tuned on a company's proprietary data and deployed behind its firewall, achieving better accuracy on that company's tasks than an enormous 100B+ parameter model trained for general use. This trend of **SLMs overtaking LLMs** is especially relevant for organizations that need to balance AI performance with confidentiality, cost, and speed of development. GO GenAI's design is influenced by this trend – embracing SLMs where appropriate to give users efficient models that can be run with stronger data control, while still interfacing with larger models when necessary for general knowledge.

Market Needs and Opportunities for AI Agents

As AI capabilities mature, there is an **explosive growth in demand for AI-driven "agents"** – software entities that can autonomously perform tasks, assist humans, and make decisions using AI models. Across industries, these agents are seen as key to automating workflows, enhancing productivity, and unlocking new services. The market indicators of this trend are striking:

- **Surging Investment:** Enterprise spending on generative AI (including agent technologies) jumped dramatically in the last two years. One analysis noted that enterprise AI spending rose from $2.3 billion in 2023 to **$13.8 billion in 2024**, as companies moved from pilot projects to production deployments[9]. AI agents are moving to the core of business strategies, not just experimental labs. Venture funding is following suit – by late 2024 the nascent *agentic AI* sector had received over $2 billion in startup funding and was valued at ~$5.2 billion[8]. Analysts project **astonishing growth** in this sector, with estimates that it could reach nearly **$200 billion by 2034**[8]. Even nearer-term forecasts are bullish: the global AI agents market is projected at **$7–7.6 billion in 2025**, on track to ~$47–52 billion by 2030 (a ~45% compound annual growth)[10][11]. In other words, organizations believe AI agents will play a substantial role in the modern economy, and they are investing accordingly.

- **Widespread Adoption:** Surveys show that **a majority of enterprises are already experimenting with or using AI agents**. NVIDIA's research cites that over half of large IT enterprises were actively using AI agents as of 2024, and 21% had adopted them *just in the last year*[8]. Another industry survey of 600 business leaders found 72% anticipate broader adoption of generative AI tools in the near future[12]. In fact, one report expects *85% of enterprises to implement AI agents by the end of 2025* as part of their operations[13]. This rapid adoption is driven by

clear use cases delivering value: for example, code-writing assistants (AI "copilots") are now in use at 51% of surveyed companies, customer support chatbots in 31%, and AI-powered search and data analysis tools in ~28%[14][15]. Businesses see these agents improving efficiency, reducing costs, and freeing up humans for higher-level work.

- **Market Use Cases:** AI agents are proving their worth in diverse scenarios. In software development, AI agents can act as junior developers – suggesting code, generating tests, and even managing parts of the DevOps pipeline. In customer service, they handle routine inquiries or assist human reps with real-time suggestions. In knowledge work, agents can sift through documents, perform research, summarize meetings, and draft reports. Even highly regulated fields like healthcare, finance, and law are finding narrow uses for AI agents (e.g. triaging patients, automating compliance checks, drafting legal documents). The broad interest across sectors indicates a *general market need for technology that can easily create and customize such agents*. Each department or domain might need an AI agent tuned to its specific tasks – and not every company will have expert AI engineers to hand-code these. This underlines the need for accessible development tools (a gap GO GenAI aims to fill).

In summary, the potential market for AI agents is enormous and rapidly expanding. Organizations are eager to deploy AI agents at scale, **provided they can do so in a way that is cost-effective, secure, and integrated with their systems**. The opportunity for a platform like GO GenAI is thus twofold: enabling more players (researchers, engineers, even non-programmers) to create useful agents, and addressing the pain points that currently slow down enterprise AI deployments.

Challenges in Deploying AI Agents in Industry

Despite the excitement and investment, deploying AI agents in real-world environments remains **challenging**. Many projects stall or under-deliver due to technical and organizational hurdles. A comprehensive state-of-the-art review must acknowledge these key challenges:

- **Data Privacy and Security:** Enterprises rank data confidentiality as a top concern when adopting AI powered by LLMs. Relying on third-party cloud APIs for AI (as many LLM-based agents do) means sensitive data might leave the organization's control. In a 2025 survey, **44% of companies cited data privacy and security as the #1 barrier to LLM adoption**[16]. Especially in regulated industries (finance, healthcare, government), sending data to an external AI service is often a non-starter. This drives the need for solutions that can run AI models on-premises or within a trusted environment, use encryption/obfuscation, or otherwise guarantee data stays protected. GO GenAI, by supporting smaller on-prem models and secure deployment options, directly addresses this concern.

- **Integration Complexity:** Integrating AI agents into existing software stacks and business processes can be technically complex. Enterprises often have legacy systems, strict compliance workflows, and specific infrastructure. Only **14%** of

organizations in one survey *initially* named integration complexity as a top challenge, but in practice many discover that *"embedding LLMs into existing systems often requires rethinking service orchestration, compliance policies, and infrastructure monitoring."*[17] In other words, plugging an AI agent into your product or workflow isn't like flipping a switch – it may require new pipelines for data, new monitoring tools for AI decisions, and new ways to enforce rules (so the agent doesn't do something undesirable). This integration hurdle is exacerbated by the fragmentation of tools: one might need a data ingestion tool, a vector database for knowledge retrieval, an ML model serving stack, separate monitoring dashboards, etc., and gluing these components together is arduous. An industry analysis noted that cross-functional AI teams benefit from a *"shared framework instead of stitching tools together"* across the AI lifecycle[18]. Simplifying integration – for example by providing a unified platform that covers most of the pipeline – is a major motivator for GO GenAI.

- **Deployment and Scalability:** Many AI projects work well in a prototype or lab setting but falter when scaled for production. Issues include latency (AI agents must respond quickly to be useful in real-time applications), concurrency (handling many users/tasks at once), and uptime (maintaining reliability). LLM-based agents can be *expensive* to run at scale – their demands for GPU compute can strain budgets. In fact, 24% of enterprises flagged high computational costs and budget limitations as a significant challenge in scaling LLM solutions[19]. Fine-tuning large models or even hosting them can cost millions. Organizations need ways to **optimize models and infrastructure**: this could mean using smaller models (SLMs) where possible, running models in quantized or optimized modes, and using efficient caching or routing (e.g. only calling the big model when absolutely needed). Another facet is *deployment optimization*: packaging an AI agent so it can be easily deployed to cloud servers, edge devices, or embedded within applications, ideally with one-click or automated pipelines. Currently, teams often have to custom-build deployment pipelines with DevOps engineering, which slows down iteration. GO GenAI's goal of "optimized deployment" is directly aimed at this complexity – making it simple and cost-efficient to go from a working agent in development to a scalable service in production.

- **Maintenance, Monitoring, and Trust:** Once an AI agent is deployed, the work isn't done. Models can **drift** in performance or make mistakes (e.g., generating incorrect or even inappropriate outputs). In enterprise settings, there must be monitoring and human-in-the-loop mechanisms to catch and correct such issues. One report found that *hallucinations* (nonsensical or false outputs) were noted as a technical issue affecting about 15% of failed AI pilot projects[20]. Ensuring **reliability** of an agent's decisions is crucial – this involves testing it thoroughly (perhaps with evaluation datasets or sandbox trials) and monitoring it continuously once live. Many organizations lack the AI-specific MLOps tooling for this, leading to trust issues. If an agent gives one bad recommendation, user trust can plummet. Moreover, business leaders often require explainability and audit logs for the agent's actions (especially under AI regulations). All these needs call for integrated **observability and governance** for AI agents (features that some

emerging platforms, like Vellum, emphasize with built-in evaluation, versioning, and audit logs[21][22]). GO GenAI will need to consider these aspects to drive adoption – providing ways to test agents, track their performance, and involve humans where needed.

- **Human Skill Gaps and Adoption Resistance:** Beyond technical issues, there is a human factor. Many engineers and domain experts are still unfamiliar with prompting AI or integrating it into products. Training staff and re-engineering processes takes time. If developing an AI agent requires deep expertise in machine learning or writing complex code, that becomes a bottleneck. Likewise, end-users or stakeholders might resist adopting AI agents if they find them confusing or untrustworthy. The **usability** of AI development tools, therefore, is paramount. Simplified programming interfaces, clear visualization of an agent's logic, and transparency in how the agent works all help broaden the pool of people who can contribute to building and using AI agents. This is a core premise of GO GenAI: making the creation of AI agents *easier to understand and manipulate*, so that more professionals (researchers, software architects, even non-coders in some cases) can engage with the technology.

In summary, deploying AI agents is not plug-and-play – companies face **integration hurdles, high costs, governance concerns, and skill gaps**. Any solution seeking to accelerate agent adoption must squarely address these issues. We see the existing state of the art attempting to do so in pieces: next we review the solutions currently available and how they meet (or fail to meet) these needs.

Existing Solutions and Competing Approaches

The booming interest in AI agents has led to a proliferation of tools and frameworks. Broadly, current solutions fall into a few categories: **developer libraries**, **visual builder platforms**, and **managed enterprise platforms**. Each approaches the problem from a different angle, and each has its advantages and limitations.

1. Code-Centric Frameworks (LLM Toolkits): These emerged first, often as open-source libraries to help developers orchestrate prompts, models, and data for building AI-driven applications. A prime example is **LangChain**, an open-source framework widely used to chain LLM calls with other functions (retrieving documents, invoking external tools, etc.). LangChain provides modular components for things like retrieval augmentation, memory, and tool integration. It's very powerful and flexible – many early agent systems were built on it – but it *"requires engineering resources for hosting, scaling, and ongoing maintenance"*[23][24]. The steep learning curve and need to write a lot of glue code means LangChain is best suited for skilled developers willing to invest time. Similarly, frameworks like **Microsoft's AutoGen** (an open-source multi-agent orchestration toolkit) enable advanced research on agents collaborating or self-reflecting[25], but they *"lack enterprise-grade governance"* and need significant custom engineering to use in production[26][27].

Other code-first offerings include OpenAI's **Functions/Agents API** (allowing developers to define "tools" that GPT-4 can use via API calls). This provides a relatively simple way to prototype agents that use OpenAI's models with function calling. However, it is

vendor-locked to OpenAI's ecosystem; while one gains ease-of-use and good safety defaults, one sacrifices model flexibility and must accept usage-based costs[28][29]. In general, purely code-centric approaches give maximum control and customizability – vital for complex or cutting-edge applications – but at the expense of higher complexity in integration and a need for expert developers. They also often require assembling multiple external systems (for database, monitoring, UI, etc.), the very fragmentation that can slow down projects.

2. Visual Programming and Low-Code Platforms: To broaden accessibility, a new wave of tools uses **graphical interfaces or low-code paradigms** to build AI agents. These tools recognize that many subject-matter experts (and even AI engineers) prefer a visual **workflow designer** over writing hundreds of lines of code or YAML configuration. One such solution is **Flowise**, an open-source platform that lets users *"build AI agents visually"* by connecting modular blocks representing data sources, model invocations, and logic[30]. Flowise supports chaining multiple agents and integrating retrieval-augmented generation (RAG) and tool usage, all through a drag-and-drop interface. It touts being *enterprise-ready* with support for over 100 models, vector databases, and on-premise deployment options[31]. Essentially, Flowise abstracts the LangChain-style chaining into a canvas where you can draw the flow of an agent's reasoning. This greatly improves **accessibility**: a developer with basic JavaScript/Python knowledge can spin up a chatbot or multi-agent workflow without mastering LangChain's code API. Another example is **Rivet**, a visual AI programming environment created by Ironclad. Rivet's interface lets teams design complex **prompt graphs** (where nodes are prompts or tool calls and edges define the flow) and debug them step by step[32]. Ironclad built it after struggling to **"build AI agents programmatically"**, finding that a visual tool *"unlocked [their] team's ability to collaborate on increasingly complex and powerful LLM prompt graphs"*[33]. This underscores how visual tools can act as a common language for teams, allowing product managers, researchers, and engineers to all contribute to an agent's design.

Crucially, visual builders often include features to ease debugging and iteration. For example, Rivet enables real-time observation of the agent's chain-of-thought and easy replay or modification of steps[34][35]. This is important because developing with LLMs is not like traditional coding – prompt engineering benefits from quick trial-and-error and **interactive tuning**, which GUIs facilitate. There are numerous other entrants in this visual arena: **Dify** (an open-source graphical agent builder with templates and self-hosting ability)[36], **Gumloop** (a lightweight drag-and-drop LLM agent prototyper)[37], and **CrewAI** (which specializes in visually orchestrating *teams* of role-based agents collaborating on tasks)[38], to name a few. Even general automation platforms like **n8n** and **Zapier** have added AI agent nodes, blending no-code app integration with AI logic[39][40].

Perhaps most significantly, the major AI/cloud providers are embracing this approach. In late 2025, Google introduced the **Google ADK Visual Agent Builder**, a browser-based IDE for designing complex multi-agent systems via drag-and-drop, natural language prompts, and integrated toolkits. As one Google developer advocate described: *"No more hand-crafting YAML. No more syntax errors. Just pure agent architecture."*[41]. The Visual Agent Builder lets you visually map out agent hierarchies, configure models and

tools through forms, and even use an AI assistant to auto-generate parts of the agent based on plain English descriptions[42][43]. It exemplifies how **graphical interfaces combined with AI assistance** can dramatically lower the barrier to creating sophisticated agents – you draw the design, and let AI handle boilerplate code. Microsoft has similarly integrated an *Agent Builder* into VS Code's AI toolkit, and OpenAI has hinted at visual flow builders for tool-using agents[44]. This validation from Google and others shows that visual/low-code agent development is not just a niche – it's becoming a standard component of the AI development toolkit.

Example of a visual AI agent workflow builder (Flowise). Blocks represent steps like searching information and summarizing results, connected in a drag-and-drop interface. Such visual tools allow users to design complex agent logic without writing heavy code, improving accessibility and speed of development.

The **advantages** of visual and low-code tools are clear: faster iteration, a gentler learning curve, and the ability for cross-functional teams to collaborate (since the "source" is a visual diagram or GUI, not a black-box script). They also often come with ready integration of common components (databases, APIs, monitoring) so you don't have to assemble those from scratch. However, there are **trade-offs**. Some visual platforms may not yet support the full flexibility or scale that expert coders might need (e.g., deeply custom logic or bleeding-edge research ideas might still require code). Additionally, using a new platform introduces its own learning overhead and potential lock-in (though many open-source options like Flowise and Dify mitigate this with self-hosting and exportable flows). For enterprise use, features like role-based access, version control of flows, and audit logs are just starting to appear in these tools.

3. Integrated Enterprise AI Agent Platforms: Recognizing the fragmentation and complexity of using multiple tools, a few companies are building **end-to-end platforms** for AI agents. These aim to combine the ease-of-use of visual builders with the robustness of production MLOps features. One example is **Vellum** (as highlighted in a recent industry report). Vellum provides a unified environment where both non-technical stakeholders and developers can *"co-build reliable, testable, observable AI agents that scale"*[22][45]. It offers a shared visual canvas, built-in evaluation harnesses, versioning, and one-click deployment to cloud or on-prem. Essentially, it tries to cover the entire lifecycle: design (visual or code hybrid), test (with automated evals and debugging traces), deploy (with monitoring and rollback). Vellum's focus on enterprise needs (RBAC security controls, audit logs, compliance guardrails[18][46]) speaks to the gap in open-source solutions which often leave those concerns to the user. Another emerging player is **IBM's Watsonx** platform, which includes tools for building and governing AI workflows (though details on agent-specific capabilities are still evolving). Even startups like **Lindy** provide managed platforms with pre-built agent templates for business processes[47][48].

The existence of these integrated platforms underscores two important points: (a) **the market sees value in unifying the AI agent tech stack**, and (b) no single solution has yet become dominant, meaning it's an open field. Each existing solution tackles part of the problem (e.g., LangChain solves chaining, Flowise solves GUI creation, Vellum solves enterprise ops), but **users still face a patchwork** when trying to go from idea to deployed agent. This is precisely the space where GO GenAI positions itself as well –

aiming to be a *comprehensive yet user-friendly solution* that covers the spectrum from model to deployment in a cohesive way.

GO GenAI: An Integrated Solution for Next-Gen AI Agents

In light of the above state-of-the-art, **GO GenAI** (Graiphic's platform) is an incredibly timely and pertinent project. It is conceived as a **unified toolchain** that addresses the key needs identified in the industry and bridges the gaps left by existing solutions. Below, we outline how GO GenAI aligns with and builds upon the current state of the art:

- **Efficiency with State-of-the-Art Models:** GO GenAI is built to leverage **state-of-the-art (SOTA) AI models** while optimizing for efficiency. This means users can incorporate the latest and greatest language models (including large models when needed), but the platform will also support **small language models** for cases where data privacy, cost, or latency make them preferable. By fully integrating SLMs (and fine-tuning capabilities) alongside APIs for big LLMs, GO GenAI offers a flexible approach: use a small on-prem model for a confidential task, but call out to a larger model for a general knowledge query if required. This heterogenous model strategy reflects best practices suggested by researchers[49][50] and ensures that the *"smaller is smarter"* trend is capitalized on. The result is an agent that can be both intelligent and efficient, without always defaulting to brute-force large models. Such efficiency also eases deployment – smaller models mean lighter GPU/CPU load, allowing on-device or edge deployments that were impossible with huge LLMs.

- **Accessible Graphical Programming:** A cornerstone of GO GenAI is its **graphical programming interface** for constructing AI agents. In spirit, it is similar to the visual builders discussed earlier (Flowise, Google ADK's builder, etc.), but GO GenAI aims to take it a step further in *usability and integration*. The interface will allow engineers and researchers to **design an agent's logic via a visual canvas** – for example, dragging nodes for actions (like "Retrieve data from source", "Call Model X for analysis", "Apply filter/logic", "Return answer to user") and connecting them to form a workflow. This eliminates the need to write complex prompt-chains in code or deal with unwieldy configuration files. As Google's example showed, a drag-and-drop approach means *"no more hand-crafting YAML"* and reduces syntax errors[41], effectively lowering the entry barrier. GO GenAI's interface will also emphasize *readability* and *transparency*: users can see at a glance what each part of the agent is doing, which is crucial for building trust in the system. By providing a **natural language-assisted configuration** (similar to ADK's AI assistant that generates agent setups from English[43]), GO GenAI could let a user simply describe the goal (e.g. "Monitor tweets about our company and alert me if sentiment is negative, with a summary") and have the platform draft an agent pipeline to accomplish it. This combination of visual design and AI-assisted development makes programming faster and accessible to those who aren't fluent in Python or ML frameworks – addressing the skill gap challenge.

- **Unified and Flexible Pipeline:** One of the greatest strengths of GO GenAI will be how it **unifies the fragmented stages of AI agent development**. Data ingestion,

model fine-tuning, prompt engineering, chaining logic, testing, and deployment – currently these often require separate tools or manual steps. GO GenAI intends to provide a *seamless flow* where you can import or connect your data, select or train a model, design the agent logic, and deploy it *all within one platform*. This end-to-end integration is a direct antidote to the "mille-feuille" stack problem the user described (layers of disparate tools making a project complex). Instead of exporting a LangChain chain to then wrap in a FastAPI server to then containerize in Docker, etc., GO GenAI would let you press a button to deploy the agent you designed, with optimal settings. As one industry commentary put it, having *"one shared framework instead of stitching tools together"* speeds up development and iteration[18]. GO GenAI also promises **flexibility in usage** – meaning it can cater to various use cases (from simple chatbots to multi-agent research systems) and integrate into different environments. Whether an academic researcher wants to prototype an experiment or an enterprise team wants to build a customer service agent, the platform should accommodate both with appropriate options. This flexibility also implies supporting multiple programming modes: a primarily visual interface for ease, but with the ability for advanced users to inject custom code or logic where needed (a "high-code when necessary" model, similar to how some teams use both LangChain and visual tools together[51]).

- **Optimized Deployment and Operations:** GO GenAI recognizes that getting an agent to **production** reliably is as important as building it. Therefore, the platform is built with deployment optimization in mind. This includes one-click or automated deployment to common targets (cloud VM, Docker/Kubernetes, or even edge devices if applicable). It will handle the packaging of models (with options like quantization or GPU/CPU targeting), set up the necessary APIs or endpoints, and ensure the agent runs with **performance monitoring** in place. By abstracting DevOps considerations, GO GenAI allows users to focus on agent behavior rather than on fiddling with servers. Moreover, drawing inspiration from enterprise platforms, GO GenAI will integrate **monitoring, logging, and feedback loops** natively. Users will be able to see execution traces of their agents (e.g., a log of each tool invocation and model response), track metrics like latency or token usage, and set up alerts or human review steps for certain conditions (a form of Human-In-The-Loop for critical tasks). These features are essential for industrial adoption because they give organizations confidence that the AI agent is **observable and controllable**. For example, if an agent in production starts outputting a lot of errors or irrelevant answers, the platform could flag this (perhaps via an evaluation metric) and even roll back to a previous stable version if needed. Such governance mechanisms map to what Vellum and others highlight as crucial for production AI (evaluation sets, version rollback, etc.)[22][52]. GO GenAI aims to bake this in from the start, distinguishing it from simpler point solutions.

- **Bridging the Adoption Gap:** Beyond technology, GO GenAI's holistic approach addresses adoption challenges. By making agent creation **intuitive** and deployment safe, it lowers the psychological and practical barriers for organizations to embrace AI agents. Engineers will not need to become AI experts

to utilize the platform – they can use familiar visual metaphors and high-level controls. This is akin to how modern web frameworks allowed many more developers to build web apps without deep knowledge of every protocol. Additionally, GO GenAI being an all-in-one platform means companies have a **single solution provider** for their GenAI agent needs, which can simplify procurement and trust. They won't need to evaluate and integrate five different tools and ensure they all work together – they can invest in one platform that guarantees synergy. In the context of the market, this positions GO GenAI as a compelling option for those 40%+ of enterprise leaders who, as surveys show, are uncertain about how exactly to implement AI but are willing to invest in it[53]. GO GenAI can offer them a clear path from idea to outcome.

In conclusion, our state-of-the-art analysis shows that **the AI agent landscape is ripe for a solution like GO GenAI**. The need for efficiency (smaller models, optimized pipelines), accessibility (graphical interfaces, low-code development), and unified deployment is well documented in both scientific literature and industry reports. Existing tools each solve part of the puzzle, but **Graiphic's GO GenAI stands out by integrating these aspects into one coherent platform**. By doing so, it directly tackles the identified pain points – it keeps data in safe environments, simplifies programming to accelerate innovation (removing the "filter" of heavy syntax), and unifies the fragmented toolchain for a smoother path to production. As companies increasingly seek to deploy AI agents at scale (a market potentially worth tens of billions of dollars in the coming years[10][54]), a platform that can reliably, securely, and quickly deliver agent solutions will be extremely important. GO GenAI's relevance is underscored by every trend discussed: it is essentially *the right solution at the right time*, aligning with both the technical evolution (SLMs, multi-agent systems) and the market demand (enterprise-ready, easy-to-use AI). By learning from and building upon the state of the art, GO GenAI has the opportunity to become a cornerstone tool for engineers, researchers, and organizations looking to harness the power of AI agents in their daily work.

References (Selected)

- Muhammad Raza, **"LLMs vs. SLMs: The Differences in Large & Small Language Models,"** *Splunk Blog*, Feb 17, 2025. Key distinctions between large language models and small language models, and their respective use cases[3][4].
- Ajay Kumar *et al.*, **"The Case for Using Small Language Models,"** *Harvard Business Review*, Sept 8, 2025. Explains why smaller specialized models may become the backbone of enterprise AI, citing NVIDIA research[6].
- Peter Belcak *et al.* (NVIDIA Research), **"Small Language Models are the Future of Agentic AI,"** arXiv preprint 2506.02153, 2025. Argues that SLMs can sufficiently power most AI agents and notes the growth of the AI agent market (>$5B in 2024, projected ~$200B by 2034)[8].
- Menlo Ventures, **"2024: The State of Generative AI in the Enterprise,"** Nov 20, 2024. Survey of 600 enterprise leaders; reports 6x increase in genAI spending (to $13.8B in 2024) and discusses challenges like implementation cost (26% of failures) and data privacy (21%)[9][55].

- TheCUBE Research (Paul Nashawaty *et al.*), **"Enterprise LLM Adoption Accelerates Amid Security and Cost Concerns,"** June 24, 2025. Finds 72% of companies plan to increase LLM spending, with data security (44%) as the top barrier. Notes integration complexity often requires rethinking infrastructure[56][17].
- Warmly.ai (Chris Miller), **"35+ Powerful AI Agents Statistics [November 2025],"** Nov 2, 2025. Collates market statistics: e.g. global AI agent market $7.6B in 2025, 45.8% CAGR to 2030 (~$47B), and ~85% enterprise adoption by 2025[10][57].
- Flowise, **"Build AI Agents Visually"** – *FlowiseAI.com* (accessed 2025). Open-source platform for visual development of agentic systems, featuring drag-and-drop workflow building and support for on-prem deployment[30][31].
- Ironclad (Rivet), **"The Open-Source Visual AI Programming Environment"** – *Rivet documentation* (2023). Describes using a visual graph interface to build and debug LLM-powered agents, facilitating team collaboration[58][33].
- Thomas Chong, **"Building AI Agents Visually with Google ADK Visual Agent Builder,"** *Google Cloud Community on Medium*, Nov 2025. Introduces Google's browser-based IDE for multi-agent systems with a graphical canvas and AI-assisted configuration – no need to write YAML manually[41][42].
- Vellum.ai, **"Top 11 AI Agent Frameworks for Developers (September 2025),"** Sept 2025. Industry blog comparing agent frameworks. Notes that LangChain is powerful but requires significant engineering effort[23][24], and highlights the value of shared visual platforms to avoid stitching together many tools[18].

---

[1] [3] [4] [5] LLMs vs. SLMs: The Differences in Large & Small Language Models | Splunk

https://www.splunk.com/en_us/blog/learn/language-models-slm-vs-llm.html

[2] [6] The Case for Using Small Language Models

https://hbr.org/2025/09/the-case-for-using-small-language-models

[7] [8] [49] [50] [54] [2506.02153] Small Language Models are the Future of Agentic AI

https://ar5iv.labs.arxiv.org/html/2506.02153

[9] [12] [14] [15] [20] [53] [55] 2024: The State of Generative AI in the Enterprise | Menlo Ventures

https://menlovc.com/2024-the-state-of-generative-ai-in-the-enterprise/

[10] [11] [13] [57] 35+ Powerful AI Agents Statistics: Adoption & Insights [November 2025]

https://www.warmly.ai/p/blog/ai-agents-statistics

[16] [17] [19] [56] Enterprise LLM Adoption Surges - theCUBE Research

https://thecuberesearch.com/enterprise-llm-adoption-accelerates-amid-security-and-cost-concerns/

[18] [21] [22] [23] [24] [25] [26] [27] [28] [29] [36] [37] [38] [39] [40] [45] [46] [47] [48] [51] [52] The Top 11 AI Agent Frameworks For Developers In September 2025

https://www.vellum.ai/blog/top-ai-agent-frameworks-for-developers

[30] [31] Flowise - Build AI Agents, Visually

https://flowiseai.com/

[32] [33] [58] Rivet

https://rivet.ironcladapp.com/

[34] [35] LangGraph Studio: The first agent IDE

https://blog.langchain.com/langgraph-studio-the-first-agent-ide/

[41] [42] [43] Building AI Agents Visually with Google ADK Visual Agent Builder | by Thomas Chong | Google Cloud - Community | Nov, 2025 | Medium

https://medium.com/google-cloud/building-ai-agents-visually-with-google-adk-visual-agent-builder-bb441e59a78c

[44] Build agents and prompts in AI Toolkit - Visual Studio Code

https://code.visualstudio.com/docs/intelligentapps/agentbuilder