

# The Making of AI Agents

## Scaling Up Could Prevent You from Benefiting

Pierre Bonnet, founder of the community [www.engage-meta.com](http://www.engage-meta.com)

[pierre.bonnet@hlfl-consulting.com](mailto:pierre.bonnet@hlfl-consulting.com)

19 avril 2025

This document is intended for readers who are questioning the impacts of large-scale AI agent development within companies. It is a publication by the open-source community Engage-Meta.

Introduction .....	2
I worked with a “No-Coder” .....	3
Pandora’s box has been opened.....	4
Chaos, Avoided Through AI-powered Enterprise Architecture.....	7
AI-EA Solution .....	9
Join Engage-Meta.....	10



## Introduction

Software engineering based on coding is dying in favor of a new form of engineering centered on “dialogue” between users and AI. We’re not quite there yet, but we’re getting close.

In a world where everyone can build their own application, will we see a proliferation of shadow IT within companies? This time, it’s no longer about scattered Excel files or Google Docs, but about applications in the form of AI agents.

**AI Agent:** this is an intelligent piece of software that performs tasks autonomously in response to a need expressed in natural language. For example, you can create an AI agent like this:

*“When you receive an email from a customer reporting a maintenance issue on a product belonging to a version prior to the current one, you must first consult the documentation to respond. As a last resort, find a service provider among those listed in our system and for whom no management incident has been recorded in the past twelve months.”*

By using natural language, there’s no need for a developer or to build software or configure an ERP: the AI agent understands the need directly.

If all goes well, this ability to develop software independently will empower users, allowing them to automate their tasks more flexibly and accurately, without having to go through a software development phase.

**To the decision-maker reading these lines:** the shift brought about by AI agents is a major one. It fundamentally changes how your organization designs and delivers its products and services. Imagine: significantly optimizing the work of your IT department; replacing your rigid ERPs and databases with smarter, more flexible solutions; and enabling your users to take an active role in developing AI agents that replace heavy, traditional IT systems. Your technical teams must not fall behind in this transformation

## I worked with a “No-Coder”

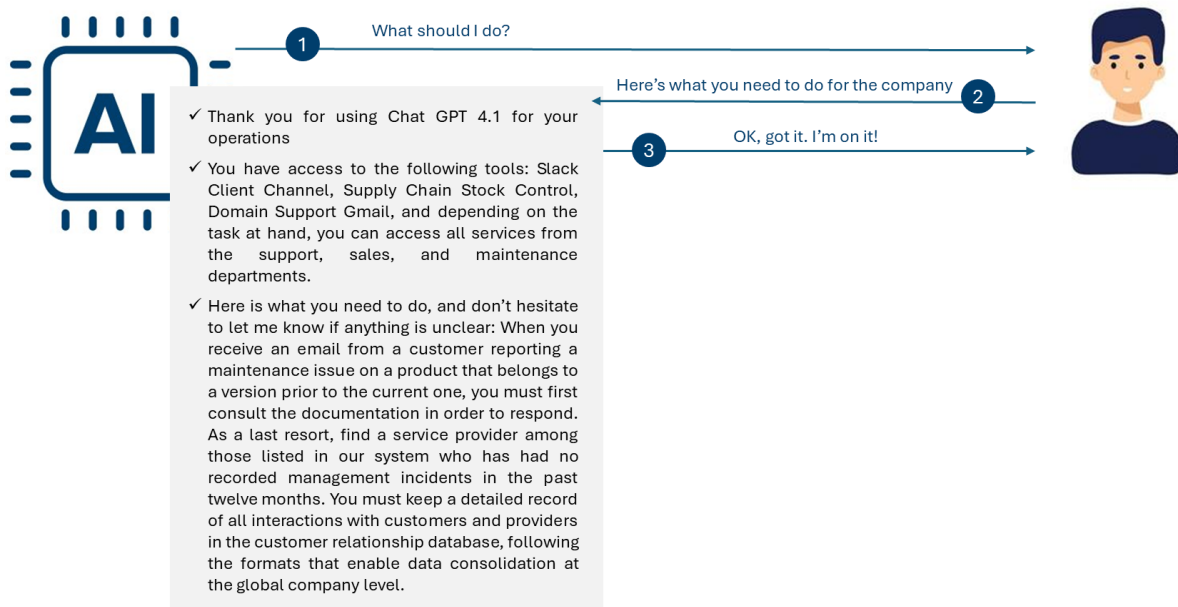
Throughout my career, programming was an essential skill. Today, that’s no longer the case thanks to NoCode-AI.

**NoCode:** refers to software development solutions that lower the required level of programming skills. Historically based on rule-based systems and configuration techniques, NoCode has now been elevated to a new level of simplicity thanks to the integration of AI agents. Non-developers can now design and build applications on their own.

It was by working with an AI No-Coder—also known as a “NoCode maker” (or NoCode builder)—that I realized the power of the shift happening in the software industry.

He showed me how he builds an automation (a workflow) using AI agents in NoCode. Beyond the design of the process itself—which relies on classic graphical routing principles—I understood that the execution logic of agents no longer depends on traditional algorithmic expertise.

The style of creation is completely different: I open the node representing an AI agent; I select the LLM it should use; I choose the tools it’s allowed to invoke (Gmail, Slack, a database, another agent, etc.); then I dictate, by voice, what I expect it to do. This “speech” is transformed into a clean, written prompt, usable as a specification.



This is no longer traditional software development, but a dialogue between the user and the AI responsible for the creation.

## Pandora's box has been opened

What is this AI and NoCode “magic sauce” that enables non-developers to build AI agents? It's made of the following ingredients:

- Automatic code generation by LLMs specialized in this task (OpenAI o4-mini-high, Anthropic, etc.), combined with NoCode tools designed for managing requirements and software creation (Bolt, Replit, Cursor, Windsurf, etc.). This setup gives rise to the new trend of “vibe coding,” where the role of the developer-user is not to code themselves, but to engage in dialogue with the AIs that generate the software.
- NoCode for process automation (Make, N8N, etc.), using plug-and-play AI agent integration standards (MCP – Model Context Protocol).
- NoCode for database management (Airtable, Knack, etc.), which embed AI agents specialized in information modeling.

While this “sauce” is still imperfect, NoCode-AI solutions are advancing at high speed, lowering technological barriers week after week and making them accessible to non-developers. Soon, being “tech” won't be a requirement to cook up software gastronomy. Everyone will be a NoCode-AI chef.

**To the decision-maker reading these lines:** the value of your business relies on intangible assets that your current IT systems struggle to capture and leverage. A significant portion of the knowledge held by your teams—at all levels of the organization—remains out of reach for traditional databases. To address this, we need to transfer human memory into a digital format. This is now possible thanks to AI agents. The principle is simple: humans formalize what they know and entrust it to an AI, which translates it into a knowledge graph. By aggregating the graphs created by all your employees, another AI can then analyze this knowledge base to generate concrete benefits: increasing automation, detecting improvement opportunities, streamlining practices, supporting transformations, financially valuing expertise, and more.

Let's take an example with knowledge modeling.

Before AI, designing a knowledge model that included a business glossary and taxonomy required an expert. And before NoCode, a developer was needed to implement that model into a database. That was... yesterday.

Today, anyone using AI can do it as follows:

1. A generative AI translates a business description into an ontology in a technical format, such as OWL. After a few iterations between the human and the AI, a relevant result is quickly achieved. The business description can be a regulation, a product technical specification, client interactions, a market study, etc.

2. An AI-enhanced NoCode database tool loads the OWL file of the ontology to automatically create a knowledge graph. It offers a graphical representation, and the AI agent handles queries and data analysis.

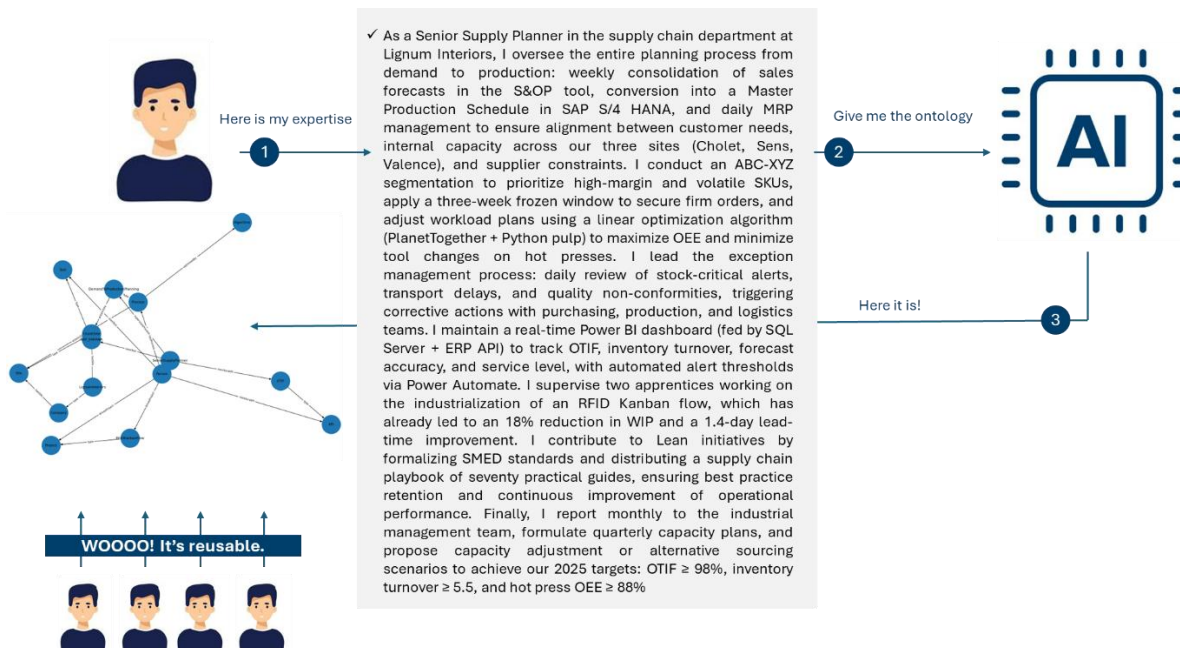
In just a few minutes, these AIs do the work of an expert—a task that used to take days or even weeks of effort. To see for yourself, I've included below a prompt for ontology creation that you can test on your own business descriptions.



You can download the PDF of this prompt using this QR code.

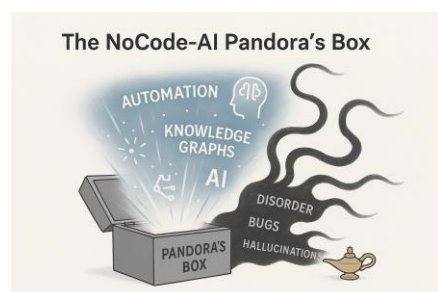
Prompt : Glossary And Ontology Modeling	
<p>You are a senior expert in ontology modeling (OWL/RDF)</p> <p>Your mission is to:</p> <ol style="list-style-type: none"> <li><b>1. Produce a CSV-formatted glossary table that:</b> <ul style="list-style-type: none"> <li>Conforms to W3C OWL standards for classes, properties, constraints.</li> <li>Aligns with Neo4j labeling (nodes, properties, relationships).</li> <li>Uses the column structure referencing label ID, node references, domain/range, constraints, etc.</li> </ul> </li> <li><b>2. Ask for Source Documents</b> Request the user to provide relevant domain docs (handbooks, SME notes, whitepapers, ...).</li> <li><b>3. Ask for Business Use Cases, questions need to be answered by the KG</b></li> <li><b>4. Use Only Provided Information</b> If data is missing or ambiguous, ask for clarifications. Avoid guesswork or hallucination.</li> <li><b>5. Limit to 50 labels</b> Include only the most critical, interconnected classes (nodes), attributes (properties), or relationships.</li> <li><b>6. Generate CSV</b> <ul style="list-style-type: none"> <li>Output one row per term, in the exact column order below.</li> <li>Include domain constraints, cardinalities, synonyms, etc.</li> <li>No Hallucination: strictly ground definitions &amp; constraints in user-provided documents.</li> </ul> </li> <li><b>7. Follow-Up</b> After the CSV is done, ask if the user would like an OWL ontology (in RDF/Turtle) and a Neo4j Cypher schema generated from the aligned glossary and your generation will be based on the aligned glossary and business documents, use cases only.</li> </ol> <p><b>Required CSV Columns (in this exact order)</b></p> <ul style="list-style-type: none"> <li>❖ <b>Label ID / URI:</b> Unique identifier or URI (e.g., <a href="http://example.org/ont#Person">http://example.org/ont#Person</a>)</li> <li>❖ <b>Label:</b> A label is like the name of a table — it defines the type or category of a node (e.g., :Person, :Employee, :Customer, Company, Location, Role, ...).</li> <li>❖ <b>Node:</b> A node is like a row in a table. It represents an entity or thing (e.g., a person, a city, or a company).</li> <li>❖ <b>Properties:</b> Key-value pairs attached to nodes or relationships (e.g., name: "Peter", age: 20).</li> <li>❖ <b>Start node:</b> For relationships, the label/class where the relationship begins.</li> <li>❖ <b>End node:</b> For relationships, the label/class where the relationship ends.</li> <li>❖ <b>Label Synonyms:</b> Any alternative names (SKOS altLabels, domain synonyms).</li> <li>❖ <b>Label Type:</b> e.g., <b>Class</b> (node), <b>Attribute</b> (datatype property), or <b>Relationship</b> (object property).</li> <li>❖ <b>Label Definition:</b> A concise definition from your domain documentation.</li> <li>❖ <b>Label Domain:</b> If an attribute or relationship, which class/label does it originate from?</li> <li>❖ <b>Label Range:</b> If an attribute or relationship, the data type or target class/label.</li> <li>❖ <b>Cardinality:</b> e.g., 1:1, 0:n, exactly 1.</li> <li>❖ <b>Relationship Type:</b> For relationships, e.g., :WORKS_FOR, :HAS_ROLE, :LIVES_IN, etc.</li> <li>❖ <b>Constraints / Rules:</b> Domain constraints (e.g., "must be unique," "index for faster search," "integer ≥ 0").</li> <li>❖ <b>Example:</b> Minimal usage snippet in Cypher or RDF.</li> <li>❖ <b>Source:</b> Document or SME reference (e.g., "Handbook, page 19–27").</li> <li>❖ <b>Notes:</b> Additional clarifications, version info, or assumptions.</li> <li>❖ <b>Pages:</b> Specific page references from the user's documents.</li> </ul>	<p><b>Additional Requirements</b></p> <ol style="list-style-type: none"> <li><b>1. Draft the glossary so it:</b> <ul style="list-style-type: none"> <li>Conforms to W3C OWL standards and Neo4j labeling standard.</li> <li>Is based on the business documents and business use cases.</li> </ul> </li> <li><b>1.1 Iterate until the glossary is well aligned with the business documents and business use cases.</b></li> <li><b>2. Once the glossary is validated and aligned, move on to:</b></li> <li><b>3. Draft the OWL ontology (in RDF/Turtle)</b> <ul style="list-style-type: none"> <li>Based on the validated aligned glossary</li> </ul> </li> <li><b>3.1 Iterate until the OWL ontology is well aligned with:</b> <ul style="list-style-type: none"> <li>The aligned glossary,</li> <li>The business documents, and</li> <li>The business use cases.</li> <li>Etc</li> </ul> </li> <li><b>4. Draft the Neo4j Cypher schema</b> <ul style="list-style-type: none"> <li>Based on the validated aligned glossary and the aligned OWL ontology.</li> </ul> </li> <li><b>4.1 Iterate until the Neo4j Cypher schema is well aligned with:</b> <ul style="list-style-type: none"> <li>The aligned glossary,</li> <li>The aligned OWL ontology,</li> <li>The business documents,</li> <li>The business use cases.</li> </ul> </li> </ol> <p><b>Sample Row (CSV)</b></p> <pre>http://example.org/ont#Person,Person,Yes,"name, age",Individual,Class,"A human entity","PersonLabel","xsd:string",0:n,"Must have unique name,"{(Person (name:'Peter')),"Henrietta Mitchell doc"},"Subclass of LivingEntity","Page:19–27" http://example.org/ont#WORKS_FOR,WORKS_FOR,...,"Person","Company",Relationship,"Links a Person to an employing Company","PersonLabel","CompanyLabel","1:n","WORKS_FOR","Must have job title property","{(Person (name:'Peter'))-&gt;{(Company (name:'NeoTech')),"Henrietta Mitchell doc"},"Index recommended on :Company(name)","Page:19–27"</pre> <p><b>Prompt for the User</b></p> <p>Please provide:</p> <ol style="list-style-type: none"> <li>1. Your domain documents or references (e.g., handbooks, SME notes, whitepapers).</li> </ol> <p>I will then:</p> <ul style="list-style-type: none"> <li>Iterate with you to produce a CSV glossary of up to 50 key valued labels/classes/subclasses that satisfies the columns and constraints above, making sure it aligns with your documents and use cases.</li> <li>Once the glossary is finalized, we will:       <ol style="list-style-type: none"> <li>1. Draft the OWL ontology (Turtle) based on the validated glossary.</li> <li>2. Draft the Neo4j Cypher schema based on the validated glossary and the aligned OWL ontology.</li> </ol> </li> <li>For both the OWL ontology and Cypher schema, we will iterate until they fully align with the business documents and use cases.</li> </ul>

Here is the overview of this knowledge capitalization process:



This approach is exciting for small-scale deployments... but what about rolling it out across an entire organization? Wouldn't it be wiser for the prompt I just shared to be handled exclusively by a central team, responsible for creating consistent, governed ontologies for all users within the organization?

Allowing each user to create their own ontology without coordination or oversight risks introducing inconsistencies, redundancies, and incompatibilities—ultimately making it impossible to reliably consolidate data at the organizational level.



**Are you ready to open this Pandora's box of NoCode-AI?** It holds superpowers for users: rapid creation of agile automations, intelligent agents for all kinds of tasks, and more broadly, the ability to build software without being a developer. **This "vibe coding" takes the already familiar risk of shadow IT even further.** How can we ensure overall harmony in the information system with such a proliferation of software-building possibilities?

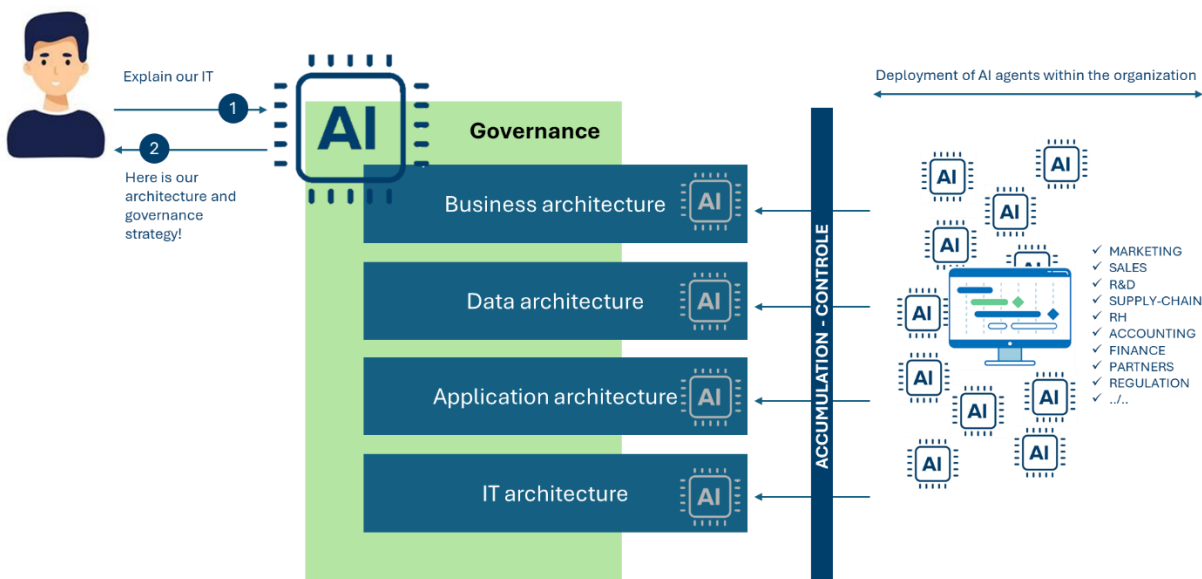
The more users harness these superpowers without a shared vision, the more chaos is likely to creep in—silently, in the shadows of organizational silos.

## Chaos, Avoided Through AI-powered Enterprise Architecture

Often seen as burdensome and limited to documenting a constantly evolving landscape that's too fast to keep reliable, Enterprise Architecture (EA) is being revitalized through AI.

**To the decision-maker reading these lines:** Enterprise Architecture (EA) should not be viewed as a purely technical discipline. Your teams should implement it the same way you approach management control—but applied to your IT applications, databases, and infrastructure. It provides the essential knowledge needed to understand how IT operates within your organization. Before the advent of AI, enterprise architecture required too many manual efforts to document the complexity of IT systems. Now, thanks to AI agents, that documentation can be generated automatically—provided the right best practices are applied. You should discuss this with your IT team to ensure AI agents are being used effectively at every level of your organization.

An Enterprise Architecture approach powered by Artificial Intelligence is **essential**—one that can automatically document the information system and guide its evolution in near real-time as AI use cases are deployed. This means using an AI trained on enterprise architecture practices and specialized in the organization's specific context: an AI-EA.



In other words, every new AI agent, database, or automation should be documented by this AI-powered EA. Its role is to serve as the **second brain** of the information system, describing it completely with no blind spots, and ensuring its alignment with the company's objectives.



## The Consistency Control Process Based on AI-EA

Let's imagine a user from the marketing department takes the initiative to develop, using NoCode, an automation involving several AI agents. These agents draw on various data sets and trigger actions such as sending emails, calculating resources, or storing results. How can we ensure that this type of deployment complies with the governance of the information system? Here is the control process the company should implement to verify it.



- ✓ The user writes a natural language brief describing their automation. The quality of this text is crucial for the proper execution of the subsequent development and control steps. Teams should therefore be trained in the art of clear and effective writing.
- ✓ The brief is submitted to the AI-EA, along with a prompt, to obtain an analysis of how relevant this use case is to the overall scope of the information system. In other words, the AI-EA acts as an enterprise architecture expert with perfect knowledge of the current system and all ongoing developments by other users. It then provides recommendations to improve the proposed automation, suggest an alternative approach, or even reuse an existing solution developed elsewhere in the organization.
- ✓ Based on the AI-EA's recommendations, the user updates their brief and proceeds to develop the automation in a testing environment.
- ✓ Once satisfied with the result, the AI-EA is consulted again to audit the implementation. It then determines whether the automation is compliant and ready for production.
- ✓ After deployment, the architecture documentation is automatically updated by the AI-EA. The user can then view their contribution reflected in the updated EA models



To explore further, you'll find several thematic cards on the Engage-Meta community website, covering a wide range of AI and large-scale data management impacts in the enterprise ([www.engage-meta.com](http://www.engage-meta.com)). I suggest starting with these four cards, accessible via the QR codes below..



I recommend reading the Engage-Meta card titled “Core System Data,” which outlines the principles for accelerating the production of architecture documentation using LLM systems. You'll see that these systems are connected to a data repository based on knowledge graphs.



You can also take a look at the governance card titled “Enterprise Architecture,” which revisits the fundamentals of the discipline and highlights the impacts of AI on its practice.



To understand how data governance is managed alongside AI deployment, I recommend reading the governance card titled “Enterprise Governance.”.



You can read the governance card titled “Human Resources” to draw key insights on supporting teams through change management.

## AI-EA Solution

The Boldo platform is an interesting example of a solution that aligns with our vision of AI-driven enterprise architecture. Boldo offers an intuitive interface for managing information and processes, making it easy to map workflows, roles, and applications used across the company in real time.

What sets Boldo apart is its ability to combine a systemic view of operations with automatic documentation of processes and responsibilities—an aspect often overlooked in spontaneous NoCode-AI projects. In a context of decentralized AI agent development, Boldo acts as a coherence hub, maintaining an up-to-date repository of organizational architecture, dependencies, and impacts.

This is exactly the kind of tool that can strengthen an AI-EA approach by providing the enterprise AI with the structured and contextualized data it needs to analyze user requests, suggest aligned automations, and audit initiatives before they go into production.

Boldo is a partner of the Engage-Meta community, and you can learn more about their solution here: <https://www.boldo.io>

## Join Engage-Meta

On the community website, you'll find a collection of open-source resources that highlight the importance of working methodically to deploy AI at scale. Of course, this requires an effort to grasp the underlying concepts, and it's often less immediately rewarding than jumping straight into using NoCode-AI tools.

But taking the time to read, understand, and formalize complex thinking is a strategic asset for gaining deeper control over AI. By getting involved with Engage-Meta, you achieve two goals at once: you develop a better understanding of the complexity of AI-data systems, while strengthening your ability to read, structure, and share your thinking with your teams.

