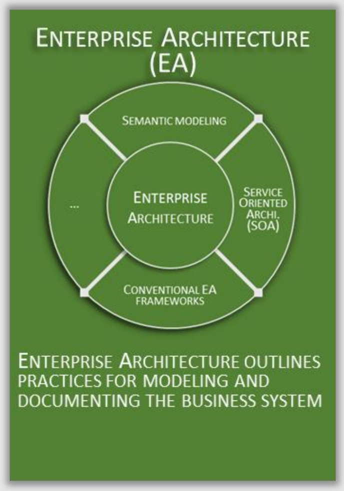
ENTERPRISE ARCHITECTURE

Enterprise Architecture (EA) outlines practices for modeling and documenting the business system. It enables the preparation and support for large-scale deployment of AI by promoting the consideration of semantic modeling (ontology) and service-oriented architecture (SOA).



1. CONDITIONS OF SUCCESS

The profitability of AI relies on the use of best practices described in TRAIDA, particularly those concerning data quality, ontology modeling, and knowledge management. Their implementation is closely linked with the company's information system, which includes the processes, rules, and data that support the execution of operations.

With TRAIDA, the goal is not to create a new AI-based system from scratch that would operate parallel to the existing one, but rather to promote a symbiosis between AI and the information system. To extend the metaphor, it's similar to the relationship between a clownfish and an anemone. Both derive benefits: the fish is immune to the stinging tentacles of the anemone, allowing it to hide there, and the anemone feeds on the fish's waste. The coupling is the same for AI and the information system. One cannot survive sustainably without the other, especially when it comes to maintaining the company's competitiveness through new information management technologies.

This coupling revolves around the value chain of the information system, which starts with the organizational processes (a) operated by the company's actors. These processes trigger rules (b) that exploit data (c). Like any chain (a-b-c), its strength depends on its weakest element. A defect in one of these intangible assets—processes (a), rules (b), or data (c)—impairs the execution of the whole. For example, an information system built around rigid silos is prone to data quality defects, which hampers the proper execution of rules. In other words, the interdependence between processes, rules, and data leaves no room for errors in any of the assets. A defect in any one of them, even minor, can have negative consequences for all the others.

Given the importance of this value chain, the integration of AI must preserve its quality. Moreover, it should contribute to greater efficiency while respecting the integrity of the three intangible assets. For example, when AI calculates the assignment of a task to an actor within an organization, the reasons behind this decision must be auditable according to the elements of the value chain:

- Processes (a): Optimize the actors' time within a general planning framework.
- Rules (b): Determine whether a treatment should be automatic, manual, or mixed, depending on the nature of the case and the regulatory context.
- Data (c): Analyze the case requiring the task to determine its nature within a predefined classification, then verify compatibility with regulatory clauses that must be adhered to.

An AI-based system that opaquely mixes several of these levels would make decision audibility and overall system maintenance difficult. In other words, each level or type of intangible asset in the information system has its own AI system dedicated to its specific concern. Of course, these levels accumulate according to automation:

- Processes: Decision-making about the steps the organization must follow to meet a need, respond to an unforeseen event, comply with a regulatory requirement (organizational level), request an actor within a team, etc.
- Rules: Automation of calculations, decision support, deduction, etc.
- Data: Analysis, compliance, consolidation, aggregation, pattern recognition, content generation, preparation of data sets for rule execution, regulatory verification (business level), etc.

Without quality data, and processes and rules that are formalized and executed with precision, it is difficult to implement intelligent and reliable algorithms whose responsibilities are clearly defined at each level. To achieve this, enterprise architecture helps improve the quality of these intangible assets within the information system:

- Instead of implementing fragmented databases with disorganized processes and rules, the goal is to streamline the quality of these assets to leverage digitization and the benefits of AI with a sufficient level of reliability and security. Good mastery of enterprise architecture helps achieve this objective.

Basic principles of enterprise architecture

Enterprise Architecture (EA) is a set of practices for classifying, modeling, and improving the quality of the intangible assets of an information system, namely the processes, rules, and data that a company uses to conduct its activities. The desired contributions are multiple, aimed at enhancing the company's operations with transparency, auditability, agility, security, regulatory compliance, reliability, and efficiency.

More concretely, EA is structured using four layers of abstraction:

1. **Business architecture**: this layer describes the processes used by the company to conduct its operations. These processes rely on steps that are automated by IT tools, as well as others that are manual. The scope of analysis for business architecture is therefore global to the company, extending beyond the IT perimeter.
2. **Data architecture**: this layer describes the architecture of the data, primarily focusing on databases. The scope of analysis is thus limited to the IT domain.
3. **Application architecture**: this layer describes the IT applications, distinguishing between different software solutions, such as custom developments and packaged software.
4. **Technical architecture**: this layer describes the technical infrastructure, including hardware, networks, and security devices.

Integration of enterprise architecture with AI

To take into account the ontologies and knowledge management required for optimal AI deployment, enterprise architecture is strengthened at two levels:

1. **The business architecture layer**: it is extended to include the consideration of ontologies to integrate data modeling at the semantic level in a unified manner, independent of databases. Thus, business architecture is no longer limited to merely analyzing processes.
2. **The data architecture layer**: in addition to the data stored in traditional databases, this layer deals with knowledge using the Enterprise Knowledge Graph (EKG) repository as described in TRAIDA. Therefore, data architecture is no longer confined to the IT perimeter; it opens up to the formalization of knowledge, independent of its level of computerization.

By adding these two components, enterprise architecture becomes an interesting framework to support large-scale AI deployment within the company, for the following three reasons:

1. **Modeling of intangible assets of the information system (processes, rules, and data)**: this modeling clarifies the value chain. It is from this chain that the best anchor points for AI-based algorithms are identified in a more efficient, reliable, and traceable manner. Without a layered structure (processes, rules, and data), the use of AI systems tends to mix organizational, business, and data processing levels, making overall governance more difficult (auditability, maintainability, impact analysis, etc.).
2. **Ontology modeling at the business architecture level**: this modeling reinforces the importance of this practice to successfully implement AI on a large scale. It is the core of the semantic platform recommended by the TRAIDA architecture.
3. **Knowledge management consideration at the same level as data architecture**: this consideration contributes to the proper integration of the Enterprise Knowledge Graph (EKG) repository.

2. IMPORTANCE OF THIS CARD FOR YOUR TRANSFORMATIVE AI

Enterprise architecture is primarily implemented with the objective of documenting the information system. Given the complexity of databases, workflows, and applications, the resulting descriptions are complex. They focus on a high level of abstraction that does not allow for concrete action to transform systems.

Worse still, since the information system is constantly evolving to meet business needs and regulatory requirements, the documentation produced by EA is rarely up to date.

Although this observation is unfavorable, the documentation effort supported by EA, even if imperfect, must be maintained; otherwise, the overall knowledge of the information system will be lost.

A more positive use of EA is nonetheless possible, especially by taking into account the specific needs for the deployment of AI systems. Indeed, adding ontologies at the business architecture layer solidifies the documentation. It makes it more sustainable than simply modeling around processes. In other words, the velocity of changes at the ontology level is lower than that observed on processes:

By clearly distinguishing the two documentation spaces (ontologies, processes), knowledge capture is more robust, easier to update, and actionable.

At the data architecture layer, knowledge management paves the way for a more powerful enterprise architecture than simply documenting databases.

This renovation of enterprise architecture is important based on your experience:

- You already have EA practice in your company, with mixed results or even a perceived failure. Rather than abandoning this documentation effort, you will take advantage of integrating AI to renovate your enterprise architecture practices. Although you consider EA as an abstract approach, distant from project needs, costly... it remains essential to better manage complexity and, therefore, contribute to better AI integration.
- You have no experience in EA and identify the need for an architectural framework to embody the information system's value chain based on processes, rules, and data. This is necessary to better control the integration points of AI-based systems in your information system.

More generally, the power of generative AI allows for the optimization of information system documentation through the following use cases:

- Automatic detection of discrepancies between documentation in the EA repository and a database or applications. Generative AI can absorb all documents, database schemas, specifications, and user guides for applications to check the overall consistency of the documentation.

- Automatic generation of documentation from descriptions of existing systems, whether they are databases, applications, data flows, regulations, security directives, user support tickets, etc. This is an opportunity to accelerate re-engineering.
- Storing EA documentation in a knowledge graph-oriented database to obtain visual representations (graphs), conduct analyses, etc. This repository functions like an Enterprise Knowledge Graph (EKG) as described in TRAIDA, but applied to knowledge about the information system.

This EKG repository, provided there is a version that describes the expected behavior of the information system, can then be used to train an AI aimed at observing the actual behavior of the information system to detect deviations from expectations. This AI is referred to as a second brain, nerve center, or trusted-AI.

CONVENTIONAL EA FRAMEWORKS

Due to their complexity, conventional enterprise architecture frameworks, such as TOGAF or Zachman, are difficult to make profitable. They are mainly used by experts, in isolation from operational teams like application designers and software developers. These frameworks are presented as repositories of best practices that cover all aspects of formalizing and governing information systems. This encyclopedic positioning does not facilitate their readability or usability.

This criticism could also apply to TRAIDA, as it is a type of enterprise architecture framework. However, since it is limited to the domain of transformative AI and data management solutions, the number of best practices remains small, which makes it easier to read and allows it to be actionable for operational teams. Conversely, because it does not cover all concerns related to the formalization and governance of information systems, it does not replace conventional frameworks.

Thus, the encyclopedic nature of EA frameworks makes them too general to provide concrete guidance, and they lack specificity to be used as a structured methodology. Although they are useful for initiating an enterprise architecture culture, it is necessary to customize them for each company's context, especially considering AI.

A major flaw of conventional enterprise architecture lies in its process-oriented approach. Indeed, the first layer of enterprise architecture emphasizes processes alone to document the information system. The next layer focuses on data; however, it is limited to logical and physical models. Consequently, the conceptual level of data is not a priority in conventional enterprise architecture frameworks.

There is a lack of conceptual data analysis, which should occur at the business architecture level, in alignment with process documentation. Without considering data early enough in the analysis, there is a risk of documenting processes without generating a positive impact on data architecture. As a result, data remains confined in silos centered around processes, leading to duplication and quality issues. This lack of data governance hinders the effective use of digitalization and AI.

A positive aspect of conventional enterprise architecture is its business focus, which aims to document processes beyond the scope of IT tools. However, as mentioned above, it is problematic that this business level is not also seen as an opportunity to model data conceptually, that is, ontologies independently of their implementation in IT databases. Worse, the next level of the data architecture layer is reduced to the IT domain, essentially databases. Yet, a wealth of information exists beyond IT systems that companies must exploit for training AI systems.

Contributions to improve EA coupled with AI

Based on these observations, conventional enterprise architecture is enhanced to serve as a facilitator for large-scale AI integration. It is then necessary to consider the four contributions presented below.

1. **Key strategic goals for enterprise architecture : t**he first contribution ensures that the enterprise architecture approach is understood and adopted by all stakeholders. It is necessary to define its objectives, including those for integrating artificial intelligence. A document titled "Key Strategic Goals for Enterprise Architecture" is prepared, containing the following four chapters:

- MOTION to clarify the objectives of the EA+AI approach.
- ENGAGEMENT to identify the tasks that need to be prioritized to achieve tangible results with EA+AI, typically within a timeframe of less than six months to avoid a tunnel effect.
- TREASURY to allocate the necessary financial resources for EA+AI operations. This chapter also describes the rules for calculating return on investment.
- ASSURANCE to outline the major governance rules for EA-AI.

To be useful to the organization, this document should be drafted in the shortest possible time, in an initial version. Ideally, within two to three weeks, stakeholders should agree on key objectives accompanied by concrete results to be delivered. It is then updated based on the progress of EA+AI in the company, usually on an annual basis.

2. **Data governance**: the second contribution is data governance with semantic modeling (ontologies). Its integration starts at the business architecture level. This step is crucial for improving data quality, a prerequisite for the effective integration of AI at the enterprise scale. This holistic approach to data analysis promotes a comprehensive understanding of the flow of information within the organization and its interactions with processes. Data governance is thus established, ensuring data quality, integrity, and accessibility. Moreover, this holistic approach ensures synergy between digitalization, AI, and business processes. It enables organizations to leverage data-driven analytics for smarter decision-making, better process automation, and ultimately greater efficiency and competitiveness.
3. **Knowledge governance**: the third contribution concerns knowledge governance. It begins with ontology modeling (as mentioned in the previous point). It now extends to the formalization of knowledge at the data architecture level. Instead of limiting the analysis to the IT scope alone, knowledge governance focuses on data that is not yet digitized, whether structured, unstructured, already formalized in writing, or tacit as individual and collective knowledge. In TRAIDA, this involves building the Enterprise Knowledge Graph (EKG), which is essential for training AI systems.
4. **AI-assisted automated governance**: the final contribution is AI-assisted automated governance. This plays a crucial role in controlling the entire information system. This mechanism is based on implementing an intelligence layer above the information system to supervise decision-making algorithms. It is known by various names, such as second brain, nerve center, or trusted AI. This AI is continuously fed with software specifications, application documentation, data structures, regulations, KPIs, etc. It observes the behavior of processes and the information system as a whole to alert on executions that do not meet expectations.

SEMANTIC MODELING

The deployment of AI systems in companies and on a large scale need to use a lot of data from the company's databases, both during their training and during prompts to enrich requests (RAG: Retrieval Augmented Generation). Since these databases and other sources such as files, archives, etc., are often heterogeneous and of varying quality, it is dangerous to connect the AIs directly to these storage areas. It is smarter to build a unified vision of all the company's data using a powerful business model that sits in front of the heterogeneous storage areas (digital twin). The AIs can then draw their data from a clean source, accompanied by security rules.

Software platforms exist for setting up this kind of system, either with a graph-oriented database approach or with the NoCode database. But regardless of the technology used, an effort of modeling is required to achieve this unified vision of the data. It also needs to be done in a way that allows for its evolution to keep up with business changes that occur regularly. Therefore, the model must be both very clear and strict in quality management, but also well-constructed enough to accept extensions without questioning everything.

This modeling involves expertise in ontology construction, also known as the art of semantic modeling. Ontology is the art of documenting the business concepts of the company and defining their relationships as well as the rules for controlling their quality.

First of all, a business concept is a key management entity for the company, such as a Client, Supplier, Invoice, Production Unit, etc. A startup has about fifteen of these, an SME more than twenty, and a large company even more. Each business concept is defined to constitute a glossary shared by the entire company. It is accompanied by a thesaurus to standardize term equivalences.

Next, the business concepts are organized into a hierarchy that describes the parent-child structures that exist between them. For example, a Client is specialized by B2B, Retail markets, etc.

Once the glossary, taxonomy, and hierarchy are in place, it is time to model the attributes of the business concepts and specify the relationships they have with each other. The semantic power of the data model greatly depends on the quality of the modeling of the relationships between business concepts. The first time you do semantic modeling, be accompanied by an expert in this discipline, at least to verify that your model is solid. You can also use an AI assistant for data modeling, but you will need to train it well before it can help you effectively.

At the end of this semantic modeling, you will have built your ontologies. At this stage, it is still a static vision of unified data. A final modeling step is needed to add a more dynamic dimension. Its purpose is to control the quality of the data contained in the business concepts. These are axioms that are added to the ontology. Here, focus on universal control rules that do not depend on organizational choices. A powerful way to formalize these business axioms is to use state machines. For example, a Product business concept could have this list of possible states: R&D, Offer Catalog, Maintenance, Out of Sale… Depending on the state of a product (instance of the Product business concept), update, delete, and usage actions are possible or not.

List of key advantages of having well-constructed ontologies

They allow the implementation of a unified data layer in front of your heterogeneous databases, or if you are starting from scratch, to have a very clean database that will follow your business evolution without creating chaos for data storage. This approach creates a digital twin of your IT on which you can plug your AIs both for their training and for prompt augmentation (RAG) by fetching real-time data in vectorized ontology instances.

They provide the necessary classification for organizing knowledge, beyond data from databases. To better train your AIs, you will need to formalize your organization's tacit knowledge, i.e., what your teams know but is not yet documented or well explained. All explicit knowledge is then loaded into the ontologies to complement structured data, thus increasing the knowledge base used by the AIs.

During prompt execution, real-time access to ontologies allows on-the-fly enrichment of the request context, enabling the AIs to work better. This is the principle of RAG.

Conversely, during the reception of the AI-generated response, access to ontologies will allow verifying the quality of the result, for example, by checking the data sources used. This significantly reduces the negative effects of hallucinations when the AI is not used in a creative context but rather for deterministic analysis.

If you are starting a business, a NoCode database with ontologies is the right way to go. If you already have an existing setup, you still need ontologies, but perhaps with a technological choice oriented towards NoCode and graph-oriented databases. Depending on the scope of your existing IT, you will need to consider the best data architecture. The TRAIDA cards will help you to decide the best choice in your context.

SERVICE ORIENTED ARCHITECTURE (SOA)

Service-Oriented Architecture (SOA) emerged in the early 1990s with the advent of Client/Server solutions. Today, it remains a valuable approach for structuring the information system around reusable services. It is reinforced by cloud platforms' microservices and DevOps engineering. It also benefits from improved implementation through databases organized by business object domains, known as data mesh architectures.

SOA is addressed at the data architecture level in the EA approach, then extends into application architecture and technical architecture. It facilitates AI implementation by clarifying the levels of responsibility in software execution (service providers and consumers) and enabling their reuse. The more sustainable the software architecture, the better AI integration benefits from good governance. Drawing a parallel with data quality, SOA acts as a powerful tool to enhance software quality, thereby improving the quality of integrating new digitalization technologies, especially AI.

To better understand this benefit, it is important to recall the essential properties that services bring with SOA architecture. A service is a process that adheres to the five properties detailed below: loose coupling, remote and interoperable activation, asynchronous operation, exposes a usage contract (interface), and complies with the SOA architecture pattern.

Property #1: Loose coupling

- A service cannot directly call another service. It delegates this function to a process specialized in chaining (orchestration).
- A service can be activated independently of its technology. To do this, activation is performed by sending (and receiving) an XML message. Therefore, it is not a binary call.
- A service can be activated in an asynchronous mode. In this case, the service subscribes to an event via an orchestration function.

Property #2: Remote and interoperable activation

- A service exposes a usage interface that is consistent regardless of its network location. The service call works regardless of the consumer's language and operating system. To promote interoperability, XML is preferred. When the service is backed by a data mesh database, the technical architecture is termed microservices. In this case, the service is reusable in a plug-and-play mode since it operates with its dedicated database for its functional scope. In practice, this type of micro-database handles a set of services around a business concept derived from ontologies.

Property #3: Asynchronous operation

- A service operates asynchronously, meaning it does not block the consumer while it executes. This principle is useful for reducing bottlenecks (performance, robustness). This type of architecture is known as Event Driven Architecture (EDA) and is combined with SOA.

Property #4: Exposes a usage contract

- A service exposes a usage contract described in two parts. The abstract part declares the input and response messages of the provided service. The concrete part describes the technical standards and protocols used for service activation. Depending on the implementation and deployment choices, there can be multiple concrete parts for the same abstract part. The usage contract is also referred to as a service interface expression.

Property #5: Complies with the SOA architecture pattern

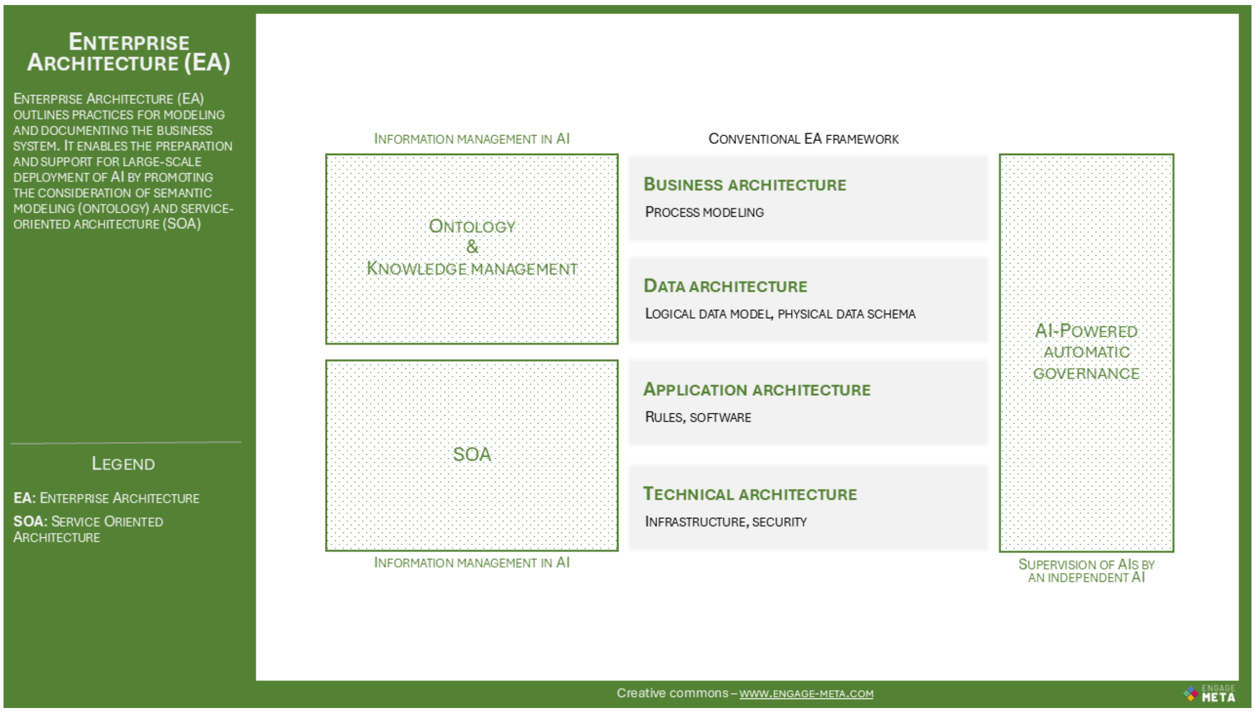
- The SOA architecture pattern involves creating an application architecture that breaks down processes into services attached to class packages. These packages form categories (business objects, concepts, or business subjects), each with an access facade that contains all the services it exposes (also referred to as a port).

Types of services

- **Business service** : this is the highest-level service in the SOA architecture. It is directly understandable by users, meaning the service providers and consumers.

- **Service exposed** by a business concept : this service is situated at the application architecture level. It represents the preferred unit for managing and reusing services. Using a "data mesh" approach allows for autonomous coupling of these services with the underlying databases that handle their business concepts.
- **Internal service to a business concept** : these are the services that implement those exposed by business concepts. These services operate at the level of detailed software engineering, particularly with components. They are not visible to users.

3. BLUEPRINT



4. YOUR SITUATION & OBJECTIVES