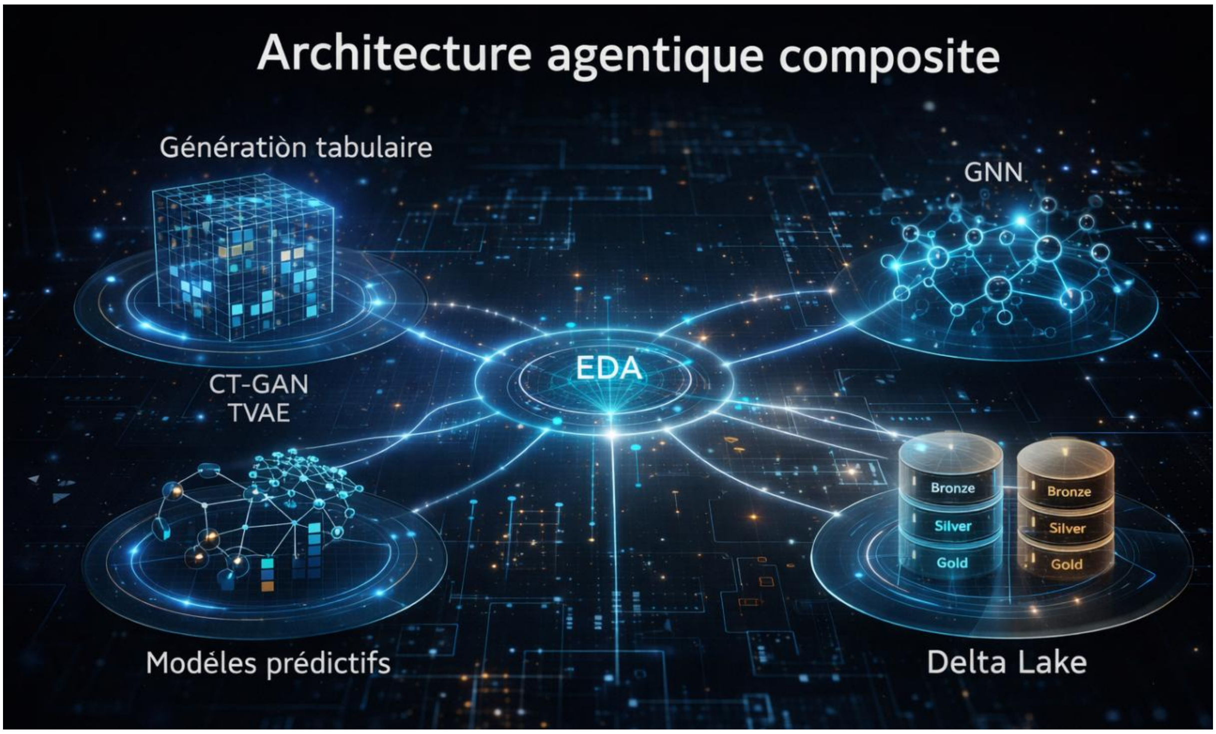

Beyond the LLM-Centric Paradigm: Composite Agentic Architecture for Digital Twins in Regulated Environments

Algorithmic Swarms, Structured Domain Memory, and Event-Driven Orchestration

Jérôme Vetillard / Twingital Institute, April 2026

Abstract

The dominant discourse on agentic AI today rests on an implicit assimilation: an agent is assumed to be a large language model (LLM) enhanced with tools, conversational memory, and orchestration mechanisms. This representation is operational for many conversational, documentary, or productivity-oriented use cases. It becomes insufficient, however, when one considers regulated environments characterized by structured tabular data, temporal dynamics, requirements for probabilistic calibration, traceability constraints, and reproducibility obligations.

This article advances a circumscribed yet firm thesis: in industrial or clinical contexts with high decision intensity, viable agentic systems generally cannot be built on an LLM as their sole computational core. They require a heterogeneous composition of specialized algorithms, a persistent and structured domain memory, and an event-driven substrate ensuring coordination, auditability, and continuous updating. Within this configuration, the LLM retains an important but delimited role: contextual interpretation, linguistic mediation, output explicitation, and interaction with humans.

We propose the term **composite agentic architecture** for this triptych composed of: (1) a set of heterogeneous specialized agents, (2) a stratified and versioned domain memory,

Jérôme Vetillard
CTO | VP R&D | Chief Product Officer | AI-Powered Healthcare & Life Sciences Products | Compliance by Design
PhD AgroParisTech | CPO MIT Sloan | Exec MBA IE Business School & Brown University
Twingital-institute / Twingital-ventures : twingital-ventures.com

and (3) an event-driven orchestration ensuring the dynamic coherence of the whole. We also introduce the concept of **functional persistence asymmetry** to designate a property observed in certain continuously ingesting digital twins: the layers of a Medallion-type lakehouse, canonically designed along an axis of qualitative data maturation, may also acquire differences in functional lifespan that, under specific conditions, allow for a structural reading in terms of working memory, episodic memory, and semantic memory in Tulving's sense [9]. This reading should be understood neither as an ontological identity nor as a universal property of the Medallion pattern; rather, it serves as a framework of architectural interpretation for a specific class of systems.

These propositions are illustrated through two application settings: the TweenMe platform and the PREDICARE/Sentinelle IA program, used here as instances of implementation rather than as self-sufficient general proof. The repositioning proposed is twofold. It is, first, epistemological: assimilating the agent to an LLM often amounts to confusing the interface layer with the computation layer. It is, second, strategic: in regulated environments, the primary need is not a model to download, but a platform for algorithmic composition capable of constructing, governing, and evolving heterogeneous systems of computation, memory, and action.

Keywords: agentic AI · composite agentic architecture · algorithmic swarm · digital twin · Delta Lake · functional persistence asymmetry · algorithmic composition · event-driven architecture · domain memory · AI Act

1. Introduction

In a previous article devoted to event-driven architecture as an essential complement to agentic AI [1], we argued that the central question was not only that of an agent's internal cognitive capabilities, but also that of its insertion into an informational, technical, and organizational environment. Event-Driven Architecture (EDA) appeared there as the substrate enabling temporal anchoring, graduated delegation, inter-system decoupling, auditability, and distributed coordination.

That initial analysis therefore focused on the **environment of action** within which the agent operates. It left open a more fundamental question: **what computational substance is an industrial agentic system actually made of?**

The contemporary market most often answers with a simple formula: an agent is an LLM capable of calling tools, reasoning in multiple steps, consulting external resources, and retaining a certain working context. This representation has genuine pragmatic effectiveness. It makes it possible to rapidly design documentary assistants, support agents, application copilots, or light planning systems. It should therefore neither be caricatured nor rejected wholesale.

Jérôme Vetillard
CTO | VP R&D | Chief Product Officer | AI-Powered Healthcare & Life Sciences Products | Compliance by Design
PhD AgroParisTech | CPO MIT Sloan | Exec MBA IE Business School & Brown University
Twingital-institute / Twingital-ventures : twingital-ventures.com

It nevertheless becomes insufficient when one turns to domains in which the agent must not only converse, but also produce, exploit, govern, or supervise specialized computations over non-linguistic data structures: clinical tabular data, physiological time series, molecular graphs, survival models, synthetic cohorts, territorial allocation constraints, risk trajectories, and medico-economic indicators. In such contexts, agentic AI can no longer be thought of as a mere extension of the conversational paradigm.

The thesis defended here is as follows: **in regulated environments with a strong tabular, temporal, and decision-intensive component, the relevant agentic system generally takes the form of a composite architecture articulating heterogeneous specialized algorithms, persistent domain memory, and event-driven orchestration.** The LLM retains a major role within such a system, but it constitutes neither its sole substance nor necessarily its computational center of gravity.

This thesis does not aim to describe all possible agents. It targets a specific class of systems, particularly digital twins operating in clinical, industrial, or territorial environments. The issue is therefore not to deny the usefulness of LLM-centric frameworks, but to identify the conditions under which their paradigm becomes structurally insufficient.

2. Terminological Clarification

A significant part of the current confusion stems from the unstable use of the term “agent.” Several analytical levels must therefore be distinguished.

2.1 Specialized Algorithmic Component

We call a **specialized algorithmic component** a model or procedure whose mathematical structure is suited to a specific class of tasks: tabular generation, supervised classification, survival analysis, graph modeling, anomaly detection, optimization, or linguistic interpretation. A CT-GAN, a Fine & Gray model, an XGBoost model, a GNN, and an LLM all fall into this category, yet they differ in both nature and guarantees.

2.2 Specialized Agent

We call a **specialized agent** a specialized algorithmic component encapsulated within a situated capacity for action. It receives certain kinds of inputs, produces certain kinds of outputs, operates under specific activation policies, publishes or consumes certain events, and interacts with a domain memory. In this sense, an agent is not necessarily an LLM. This terminological choice deliberately de-psychologizes the notion of agency: what matters is not an appearance of conversation or intentionality, but a function situated within a system of transitions, states, and dependencies.

2.3 Composite Agentic System

We call a **composite agentic system** the coordinated ensemble of several heterogeneous specialized agents, coupled to a shared memory and a coordination infrastructure. It is at this level that one may meaningfully speak of agentic architecture in the strong sense.

2.4 Digital Twin

Finally, we call a **digital twin** a particular class of composite systems aimed at maintaining an operational, evolving, and exploitable representation of a real referent, whether individual or collective: patient, cohort, molecule, infrastructure, or territory. Not all agents are therefore digital twins. However, some advanced digital twins may be understood as composite agentic systems.

This clarification avoids two symmetrical errors: reducing every agent to an tooled chatbot, or decreeing that any worthy agent must already be a fully fledged digital twin.

3. The LLM-Centric Bias: Origins and Scope

The contemporary assimilation between agent and LLM is not the result of a theoretical demonstration. It stems rather from a historical conjunction of new technical capabilities, market demonstration logic, and the conceptual legacy of the chatbot.

First, LLMs produced a legitimate cognitive shock. Their ability to summarize, reformulate, plan, code, question, and synthesize created an impression of functional generality. That impression is not entirely illusory, but it has often been extrapolated beyond its domain of validity. The confusion between the ability to **talk about a domain** and the ability to **compute within a domain** is one of the main drivers of the LLM-centric bias.

Second, ease of prototyping played a decisive role. It is possible to build, in a matter of hours, a conversational agent capable of chaining multiple tool calls and producing a convincing demonstration. By contrast, constructing a validated pipeline for tabular generation, calibrated predictive modeling, or counterfactual simulation requires far longer development, validation, and documentation cycles. The market has therefore privileged architectures that maximize **time-to-demo**, sometimes conflating demonstrative maturity with architectural maturity.

Third, the concept of the agent has been reinterpreted through the chatbot paradigm. Instead of conceiving the agent as a system situated within event flows, endowed with states, action policies, memory dependencies, and heterogeneous capabilities, it has often been imagined as an interlocutor made more autonomous. This translation is understandable. It is not neutral. It has led to an overvaluation of discursive coherence as the primary criterion of operational intelligence.

The concept of the agent, already well established in the multi-agent literature [3], has thus been re-centered around the LLM as a supposedly universal cognitive core. Contemporary frameworks have crystallized this identification by proposing an abstraction in which “agent” often means, in practice, “an LLM that decides which tool to call.” These factors explain the dominance of the paradigm. They do not demonstrate its universality.

4. Limits of the LLM-Centric Paradigm in Regulated Environments with Strong Structured Components

The point here is not to argue that an LLM is incapable of contributing to complex systems. It is to show that, in certain contexts, it does not by itself constitute the appropriate computational support for the critical functions upon which the operational validity of the system depends.

4.1 Tabular Generation and Structural Fidelity

An LLM can describe a dataset, comment on descriptive statistics, suggest transformations, and even steer an external synthetic generation tool. By contrast, the production of tabular synthetic cohorts faithful to a real distribution relies on specialized mechanisms distinct from those of language models. It involves modeling joint distributions, conditional dependencies, interactions between categorical and continuous variables, and appropriate validation protocols.

In this domain, the issue is not merely to generate plausible examples, but to preserve, in a controlled manner, statistical properties that are relevant for downstream tasks. This presupposes dedicated approaches such as CT-GAN [4], TVAE, Gaussian copulas, or other specialized architectures. The important point, therefore, is not that an LLM could never participate in such a pipeline, but that it constitutes neither its central mathematical mechanism nor its main source of validity.

4.2 Supervised Prediction on Tabular and Clinical Data

In many structured problems, especially clinical or medico-economic ones, the best operational performance is still often achieved by specialized model families: gradient boosting, forests, survival models, dedicated temporal architectures, or graph-based models depending on the case. The comparative literature continues to evolve, but the available evidence makes it difficult to maintain that an LLM, taken as the primary engine, would currently represent the best general choice for typical tabular tasks [5].

Moreover, the decisive issue is not merely raw performance. It also concerns the ability to specify variables clearly, control feature sets, inspect model behavior, document training pipelines, and produce outputs compatible with clinical or regulatory evaluation procedures. In that context, asking an LLM to bear an entire structured prediction task alone often amounts to mobilizing a capacity for linguistic mediation where the task primarily requires explicit statistical machinery.

4.3 Calibration, Uncertainty, and Regulatory Usability

In a regulated environment, a prediction is useful only if its uncertainty regime can be characterized, tested, and documented. Operational requirements concern not the discursive fluency of the system, but the quality of its measurable outputs: calibration, discrimination, stability, sensitivity, specificity, robustness, drift, and reproducibility.

LLMs may contribute to explaining uncertainty or mediating probabilistic outputs computed elsewhere. They should not, however, be conflated with the source of probabilities presumed to be calibrated and auditable. In other words, they may **interpret** a confidence regime, but should not be presumed to be its statistical foundation merely by virtue of their interface role.

4.4 Traceability and Governance of Transformations

Within a framework governed by requirements of documentation, traceability, and update management, the architectural problem does not reduce to invoking a performant model. It also involves the provenance of the data, the rules of transformation, the versioning of models, the documentation of design choices, validation procedures, and the ability to reconstruct a decision chain.

At this level, the center of gravity of the system shifts toward algorithmic composition, domain memory, and orchestration. Contemporary normative frameworks, whether the European AI Act [6] or risk management frameworks such as the NIST AI RMF [7], do not prescribe a specific architecture. They do, however, make certain architectural properties more plausible and governable than others, especially in matters of logging, human oversight, documentation, and control over transformations.

5. Toward a Composite Agentic Architecture

5.1 General Principle

A composite agentic architecture rests on three inseparable elements: a set of heterogeneous specialized agents, a persistent domain memory, and an event-driven orchestration ensuring the dynamic coherence of the whole.

This structure shifts the problem away from prompt orchestration toward composition engineering. The core difficulty is no longer simply to make a model reason in multiple steps, but to make different classes of computation cooperate, operating on different objects, with different guarantees, within a persistence and traceability regime compatible with the requirements of the domain.

5.2 Real Rather than Merely Narrative Heterogeneity

A distinction must be made between two forms of heterogeneity.

In purely LLM-based multi-agent architectures, heterogeneity is often primarily narrative: several instances of the same computational substrate are specialized through prompts, roles, or instructions. Such an approach has value. It enables a division of conversational or symbolic labor.

In a composite architecture in the strong sense, heterogeneity is **mathematical and computational**. Components differ not merely by instruction, but by the nature of the problems they address and by the guarantees they can provide. A tabular generator, a survival model, a graph neural network, an optimization engine, a calibration module, and an LLM do not occupy rhetorically different roles on the same substrate. They belong to distinct computational regimes.

5.3 Example of Functional Allocation

In a healthcare digital twin context, several families of specialized agents may be distinguished.

Generative agents are tasked with producing synthetic populations that preserve, to an evaluated extent, certain statistical properties of the real cohort. Architectures such as CT-GAN or TVAE fulfill this function.

Predictive agents are tasked with producing calibrated predictions on clinical or organizational outcome variables. They may rely on gradient boosting models, survival models, Bayesian models, or other approaches adapted to the problem.

Specialized deep learning agents operate on structures that language does not exhaust: molecular graphs, complex time series, constrained representations, and so forth. GNNs in molecular toxicology or certain families of clinically informed networks belong to this level.

Finally, an **interpretation and interface agent**, often based on an LLM, consumes the structured outputs of the preceding agents in order to make them intelligible to human users, contextualize alerts, reformulate results, explicate limitations, or adapt the register of expression to the intended audience.

The central point here is functional: the LLM does not disappear, but it ceases to be presumed to constitute the system's computational engine by itself.

6. Domain Memory: Beyond Conversational Context

6.1 A Swarm Without Memory Is Merely a Processing Chain

The specialized agents described above cannot function in isolation. A synthetic generator presupposes a source cohort. A predictive model presupposes consolidated and versioned states. A linguistic interpreter presupposes a context of trajectory, history, or comparison.

Composite architecture therefore requires a **domain memory**, understood not as mere data storage, but as the set of structures through which a system preserves, makes accessible, transforms, and reuses relevant information over time.

At a minimum, this memory includes raw data and timestamps, consolidated states, constructed features, intermediate outputs, trained models and their versions, validation artifacts, decision traces, provenance metadata, and quality information.

6.2 The Canonical Medallion Pattern: An Axis of Qualitative Maturation

The Medallion architecture of the Bronze / Silver / Gold type canonically rests on an axis of qualitative data maturation: raw, cleaned, exploitation-oriented. This axis is not, in itself, a temporal one. In a conventional analytical warehouse, one may perfectly well have very recent Gold and very old Bronze. One must therefore avoid the naive mistake of immediately equating Medallion with temporal memory.

6.3 Functional Persistence Asymmetry

In a more specific class of systems, particularly continuously ingesting digital twins, an additional property may nevertheless appear. We propose to call **functional persistence asymmetry** the difference in operational lifespan between layers, a difference that then partially overlaps with the canonical quality axis.

Bronze receives raw signals, events, and elementary observations. These objects retain durable technical utility for audit, replay, or reconciliation. Their **direct cognitive function** in the day-to-day functioning of the twin is, however, often brief: once consolidated, they generally cease to be the principal units of decision.

Silver corresponds to consolidated, cleaned, structured states that can be compared with one another. These objects remain operational for longer. They support profile updates, feature recalculations, trajectory reassessments, and longitudinal comparisons.

Gold gathers more stabilized artifacts: validated cohorts, versioned models, consolidated indicators, calibrated thresholds, and domain-derived knowledge. These objects generally have the greatest functional persistence. They accumulate, are revised slowly, and constitute the basis of the system's long-term memory.

We do not claim that this asymmetry is universal. We argue that it appears in certain systems meeting at least three conditions: continuous ingestion, a persistent object whose identity accumulates over time, and a marked difference in functional lifespan between layers.

6.4 Structural Reading in Terms of Memory

When these conditions are met, a structural reading may be proposed based on the work of Tulving [9] and Baddeley [10].

The Bronze layer may be understood as occupying, within the cognitive economy of the system, a dominant function akin to **working memory** or a processing buffer: strong proximity to perception, immediate utility, and short direct functional lifespan despite technical persistence for audit.

The Silver layer may be read as a form of **operational episodic memory**: it maintains consolidated, temporally contextualized states that can be used to follow trajectories and reconstruct sequences of transformation.

The Gold layer may be understood as a **domain semantic memory**: it crystallizes regularities, models, thresholds, and bodies of knowledge that are relatively decontextualized from the raw episode of acquisition.

This correspondence should not be read as an ontological identity between cognitive psychology and data architecture. It should be read as a **functional homology** concerning the dominant role of the layers within the economy of the system. That distinction matters. People are strangely fond of confusing analogy with proof, and then act surprised when their metaphors start biting.

6.5 Architectural Consequence

In systems where this asymmetry is indeed present, a significant architectural economy emerges. The Medallion pattern no longer serves only to organize data quality; it also becomes the main support of the twin's memory. In other words, the qualitative maturation of data and its progressive crystallization into knowledge may, under certain conditions, be handled by the same architectural device viewed from two different angles.

This proposition does not exempt us from thinking about complementary mechanisms of consolidation, selective forgetting, hierarchical organization, or governance of representations. It merely indicates that it is not always necessary to add an entirely separate memory layer in order to obtain an operational domain memory.

6.6 Difference from an LLM's Session Memory

The difference from an LLM's context window is qualitative. Conversational context is a bounded, linear, essentially textual session memory. Domain memory is stratified, multimodal, versioned, persistent, reinscribable, and tied to policies of quality, security, and audit. The former serves interaction. The latter sustains the operational continuity of the system.

7. Event-Driven Orchestration as a Logic of Circulation

Composite architecture cannot function without a coordination mechanism. That is the function of the event-driven substrate.

Events play several roles here. They trigger processing. They connect components without excessive coupling. They make system transitions visible. They ensure temporal traceability. They enable graduated reactivity according to signal type, level of criticality, and delegation policy.

Within a digital twin, a raw event may feed the Bronze layer. Its consolidation may produce an updated Silver state. That state may in turn trigger a reassessment of risk, the updating of a Gold artifact, and potentially the emission of an alert interpreted by an LLM or transmitted to a human operator. EDA is therefore not merely a communication bus. It constitutes the logic by which states, triggers, and decisions circulate within a composite agentic system.

The triptych thus becomes clear: heterogeneous specialized agents operating over structured memory, coordinated through an event-driven substrate. It is this articulation that makes a genuinely operational digital twin possible.

8. Why the Download-and-Deploy Paradigm Is Insufficient

A large portion of the market operates according to a simple logic: select a pre-trained model, fine-tune it or wrap it, and then integrate it into an orchestration framework. This logic works in many cases. It becomes insufficient when the issue is no longer access to a model, but the **validated composition of a heterogeneous computational system**.

The critical components of a composite architecture are often not general-purpose foundation models that can simply be imported as they stand. They must be built, parameterized, validated, and documented on the basis of domain-specific distributions, data schemas, business constraints, and evaluation requirements.

The primary challenge is therefore no longer to choose a model, but to ensure the compatibility, chaining, cross-validation, versioning, and traceability of several classes of models within a single operational platform.

This is what we call here a problem of **algorithmic composition**.

9. Illustrative Settings: TweenMe and PREDICARE

The foregoing considerations may be illustrated by two systems drawn from our own work. They are used here as **implementation settings**, not as self-sufficient general demonstrations of the thesis.

9.1 TweenMe as a Composition Platform

TweenMe may be described as a digital twin generation platform grounded in the composition of specialized algorithms calibrated on domain data. Its general pipeline includes cohort ingestion and profiling, synthetic generation, statistical validation, predictive model construction, an interpretation layer, and deployment on a coordination infrastructure.

In the context of the OCTOPUS oncology study, the pipeline achieved a high TSTR score under the selected protocol. Such a result is not, by itself, sufficient to conclude general statistical indistinguishability between the real cohort and the synthetic cohort. It does, however, constitute a strong indication of **operational fidelity** for the downstream tasks considered within that evaluation framework. For the purposes of this discussion, the important point is not the celebration of an isolated metric, but the fact that this kind of result rests on a chain of composition and validation rather than on the download of a single model.

9.2 PREDICARE as Territorial Instantiation

The PREDICARE/Sentinelle IA program, for its part, illustrates composite architecture at the scale of a healthcare territory. The central idea is not to deploy a single predictive model, but to organize several specialized twins or sub-systems oriented toward different functions: individual evolution, demographic dynamics, territorial allocation, event anticipation, and medico-economic efficiency.

This setting is not invoked here as exhaustive proof of final clinical effectiveness. It is mobilized as an **architectural materialization** of the thesis: such a system cannot be thought of as an enriched chatbot. It requires a coupling between specialized computation, domain memory, and event-driven orchestration.

10. Discussion and Limitations

Several precautions are necessary.

First, this thesis does not apply to all agents. There are classes of agents for which the LLM may legitimately remain the central component: documentary assistance, conversational support, office workflows, software mediation, or light task coordination.

Second, the opposition between LLMs and specialized models must not be absolutized. In many real systems, the LLM may act as a control interface, a secondary supervision layer, an inter-model translation mechanism, or a support for explicitation. The issue is therefore not exclusion, but proper localization within the architecture.

Third, the notion of functional persistence asymmetry deserves further theoretical and empirical development. It should be tested on other configurations besides those that inspired it, and it should not be mechanically extended to every lakehouse or every digital twin.

Fourth, the structural reading inspired by Tulving must be handled with caution. It should not be understood as a naive importation of a psychology of human memory into data architecture. It proposes a functional interpretive framework whose interest lies in its explanatory capacity, not in a claim of strong identity between the two domains.

Fifth, the regulatory argument must not be oversimplified. Normative frameworks do not mechanically impose a given architecture. They do, however, make some architectures far more plausible than others in terms of compliance, validation, and auditability.

Finally, the field is evolving rapidly. Foundation model capabilities on certain structured tasks are improving. The thesis defended here concerns the current state of the problem and the architectural properties required in the environments under consideration. Even if the computational center of gravity of certain subsystems were to evolve, the need for persistent domain memory and event-driven orchestration would remain.

11. Conclusion

The center of gravity in the evaluation of an agentic system is shifting. Within an LLM-centric paradigm, the quality of an agent tends to be measured by the verbal coherence of its reasoning: its ability to decompose a problem, plan, and respond credibly. Within a composite architecture, evaluation shifts toward the measurable quality of computation, memory, and transitions: fidelity of synthetic cohorts for the uses considered, calibration of predictions, robustness of transformation chains, quality of domain memory, auditability, reproducibility, and organizational integrability.

The unmet strategic need is therefore that of a platform that does not begin with an LLM as its design premise, but is capable of ending with an LLM when linguistic mediation becomes useful. Such a platform takes structured domain data as input and produces, as output, a heterogeneous system of computation, memory, and action deployable on an event-driven substrate.

In regulated environments with strong tabular, temporal, and decision-intensive components, useful agentic AI cannot be reduced to the orchestration of a language model. It tends instead to take the form of a composite architecture articulating specialized computation, persistent domain memory, and event-driven coordination. In such environments, the agent is not an interlocutor made autonomous. It is a system of computation, memory, and action. Such a system is not downloaded. It is composed.

References

[1] J. Vetillard, "Event-Driven Architecture as an Essential Complement to Agentic AI: From Delegated Cognition to Situated, Traceable, and Governable Action in Enterprise Environments," Twingital Institute, 2026.

[2] F. P. Brooks, *The Mythical Man-Month: Essays on Software Engineering*, Addison-Wesley, 1975.

[3] M. Wooldridge and N. R. Jennings, "Intelligent Agents: Theory and Practice," *The Knowledge Engineering Review*, vol. 10, no. 2, pp. 115-152, 1995.

[4] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling Tabular Data Using Conditional GAN," *Advances in Neural Information Processing Systems*, 2019.

[5] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why Do Tree-Based Models Still Outperform Deep Learning on Typical Tabular Data?" *Advances in Neural Information Processing Systems*, 2022.

[6] Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonized rules on artificial intelligence.

[7] National Institute of Standards and Technology, "Artificial Intelligence Risk Management Framework (AI RMF 1.0)," NIST AI 100-1, 2023.

[8] J. Vetillard, "Clinically-Informed Neural Networks," working doctrinal note, Twingital Institute, 2025.

[9] E. Tulving, "Episodic and Semantic Memory," in *Organization of Memory*, Academic Press, 1972, pp. 381-403; E. Tulving, "How Many Memory Systems Are There?" *American Psychologist*, vol. 40, no. 4, pp. 385-398, 1985.

[10] A. D. Baddeley, "Working Memory," *Science*, vol. 255, no. 5044, pp. 556-559, 1992; A. D. Baddeley, "The Episodic Buffer: A New Component of Working Memory?" *Trends in Cognitive Sciences*, vol. 4, no. 11, pp. 417-423, 2000.

Jérôme Vetillard
CTO | VP R&D | Chief Product Officer | AI-Powered Healthcare & Life Sciences Products | Compliance by Design
PhD AgroParisTech | CPO MIT Sloan | Exec MBA IE Business School & Brown University
Twingital-institute / Twingital-ventures : twingital-ventures.com