Universal Library of Engineering Technology
Research Article

ISSN: 3064-996X | Volume 3, Issue 3

Open Access | PP: 13-18

DOI: https://doi.org/10.70315/uloap.ulete.2026.0303003

Comparative Analysis of Approaches to Embedding AI Functionality through In-Application Interfaces and Autonomous Agent Orchestrators in CRM Systems

Bhanu Chand Somarajpalli

Sr Salesforce Cloud Architect, Baltimore, USA.
ORCID: 0009-0001-8023-8638

Abstract

Customer relationship management platforms now embed artificial intelligence in two structurally different ways, and the choice between them shapes cost, governance, and the pace of organizational change. The first way places assistive models within the interfaces teams already use, so that scoring, summarization, and drafting occur at the point where work is performed. The second way introduces a separate orchestration tier in which semi-autonomous agents pursue goals across several systems while users remain in familiar screens. This review examines the two approaches through a seven-layer reference architecture that places each in a specific tier and clarifies why the boundary between them lies between the model and orchestration tiers. The analysis treats both as design options with measurable trade-offs across change management, expressiveness for multi-step work, architectural coupling, and the surface available for monitoring. Evidence from enterprise practice and the research literature suggests that embedded assistance is more successful during first-generation adoption in mature organizations because it leverages existing processes, whereas a dedicated agent tier yields stronger long-term outcomes for second-wave programs that target genuine cross-system automation. Supporting comparisons across data access, integration style, the trust model, and migration strategy show that the central decision sits inside a wider system of architectural commitments that the framework makes explicit for practicing architects.

Keywords: *Customer Relationship Management; Embedded AI; Agent Orchestration; Enterprise Architecture; Human-In-The-Loop; Data Grounding; Integration; Salesforce.*

INTRODUCTION

Customer relationship management has become one of the most active areas of applied artificial intelligence, as the data in a CRM platform already describes customers, cases, opportunities, and the history of every interaction (Ledro et al., 2022). The integration of artificial intelligence into CRM holds substantial potential to improve organizational effectiveness, yet the managerial and architectural elements that enable successful integration remain only partially understood (Ledro et al., 2023). As model capability has grown, the question facing platform owners has shifted from whether to adopt AI to where AI should live within an existing landscape of clouds, flows, and legacy back ends.

Two patterns of placement now dominate enterprise practice, and they differ more at the architectural level than at the feature level. The first pattern embeds assistive models into the interfaces and process steps that staff already use, so that a console receives reply suggestions and a flow receives a prompt that drafts text. The second pattern establishes a distinct orchestration tier in which an agent receives a goal, selects tools, and coordinates actions across multiple systems while the human continues to work on the original screen. Both deliver value, and both appear under the same product banners, which is part of why teams often blur them together.

The blurring carries a cost, because the two patterns solve different classes of problems, traverse different layers of the platform, and impose different demands on change management and governance. When an organization reaches for embedded assistance to solve a problem spanning several systems, the orchestration logic ends up in prompts

Citation: Bhanu Chand Somarajpalli, "Comparative Analysis of Approaches to Embedding AI Functionality through In-Application Interfaces and Autonomous Agent Orchestrators in CRM Systems", Universal Library of Engineering Technology, 2026; 3(3): 13-18. DOI: https://doi.org/10.70315/uloap.ulete.2026.0303003.

or oversized flows, making it hard to reason about. When an organization stands up an agent tier for a task that a single inline suggestion would have handled, the design effort and the organizational negotiation exceed the value returned. This review addresses that boundary by comparing Approach A, which embeds assistive models within existing interfaces and flows, with Approach B, which introduces an autonomous agent tier as a separate orchestration layer. The two are examined through a seven-layer reference architecture that names the tiers of a modern CRM AI platform and shows where each approach sits, so that architects gain a structured way to decide where capability belongs, to understand the trade-offs that follow, and to see how the decision connects to neighboring choices about data access, integration, trust, and data migration.

METHOD

A useful way to reason about AI placement in CRM is to treat the platform as a stack of layers, where each layer has a defined responsibility and the boundaries between layers mark the points at which design decisions become consequential. Practice across enterprise deployments has converged on a layered pattern that runs from the channels through which people interact, down through business applications, an AI tier, a trust tier, a data tier, an integration tier, and finally the legacy systems of record, a framing consistent with the broader view that enterprise architecture itself functions as a dynamic capability for scaling and governing generative AI inside large organizations (Ettinger, 2025).

The top layer carries the channels and user interfaces through which customers and staff reach the platform, including web and mobile front ends, portals, collaboration tools, chat, and email. Beneath it sits the business application and process layer, which holds the core clouds for sales, service, and marketing together with the declarative and programmatic logic that encodes how the organization works. The AI orchestration layer comes next, and it contains both the prompt-driven and model-driven assistance that operates inside processes and the agents that coordinate work across them, while below it a trust and security layer mediates every model interaction, enforces policy, and provides a consistent governance surface.

The lower half of the stack concerns data and connectivity. The data and grounding layer unifies CRM records, knowledge content, and a customer data platform so that models can be grounded in current, governed information. The integration layer provides the connectors, gateways, and event backbone that link the platform to systems beyond it, and the legacy layer holds the enterprise resource planning, billing, claims, mainframe, and on-premises databases that remain the authoritative sources for much of the business. Figure 1 presents this reference architecture and marks where the two approaches sit within it.

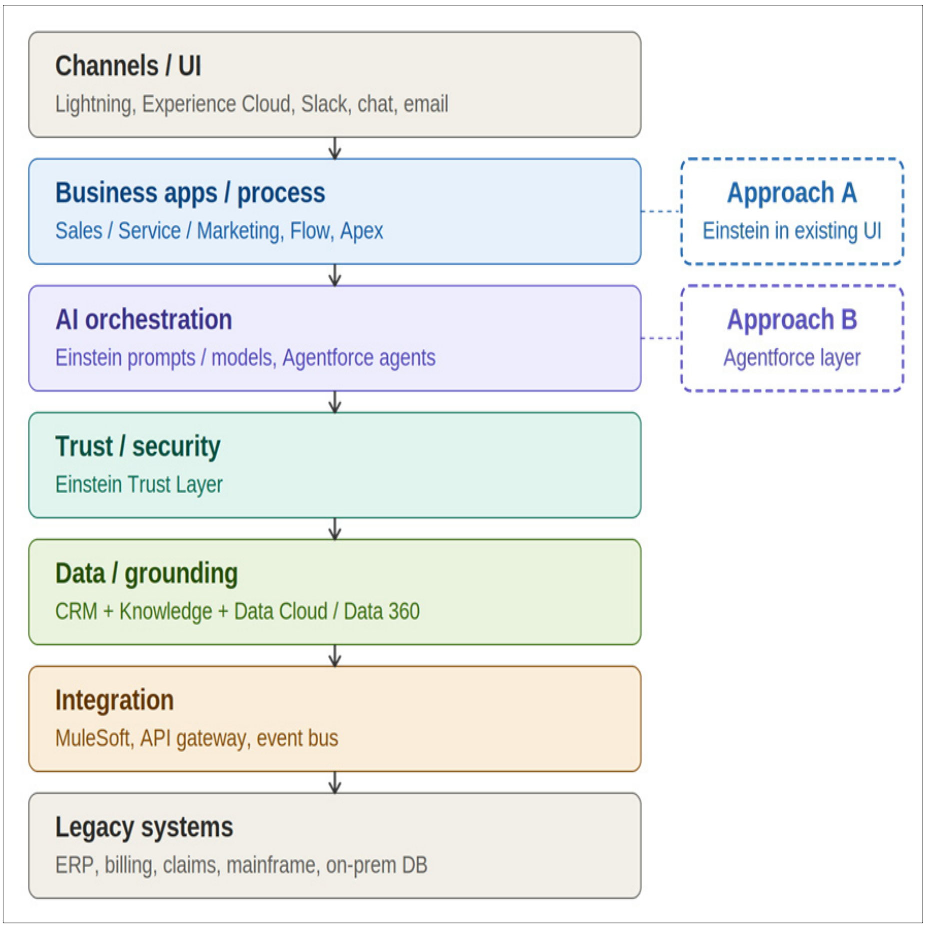

Fig. 1. Seven-layer reference architecture for CRM AI platforms with the placement of Approach A and Approach B

The figure makes the central boundary visible. Approach A operates across the channel and business application layers, since its assistance lives where staff already work and attaches to processes that already exist, while Approach B occupies the AI orchestration layer as a distinct tier whose agents sit between the applications above and the integration and data services below. The line that separates the two approaches runs through the AI layer itself, dividing model assistance that augments a single step from orchestration that drives a multi-step task. Approach A is the embedding of model capability into established interfaces and flows, where the unit of value is a faster step inside a process that a person still drives, while Approach B is the introduction of a semi-autonomous worker with a scoped goal, a set of tools, and explicit guardrails that carries a task across systems while the person supervises. The criteria used to compare them are the effort and risk of change management, the expressiveness available for multi-step work, the degree of architectural coupling, and the clarity of the surface available for governance.

RESULTS

Each approach carries an internal logic that explains both its strengths and its characteristic failure mode, and the discussion below treats Approach A first, then Approach B, then sets them against each other, and finally widens the frame to the surrounding architectural choices.

Approach A: Embedded Model Assistance

Approach A places assistance at the point of work. Typical instances include reply suggestions inside a service console, automatic summaries of calls and cases, propensity scoring for leads and opportunities, prompt-driven construction of flows, and retrieval-grounded answers surfaced within records. Each of these augments a step that a person already performs, and each leaves the surrounding process intact. The appeal of conversational and assistive interfaces in service settings is well documented, since a positive experience with such interfaces correlates with measurable gains in customer outcomes when the assistance is designed with care (Nicolescu & Tudorache, 2022).

The conditions under which Approach A performs well follow from its nature. It suits situations where an organization wants a low-friction uplift inside processes that staff have already adopted, and it suits processes whose boundaries are well defined and contained within a single team, where the work does not yet need to reach across functions, so that the assistance amplifies established behavior at low cost.

The observed advantages cluster around adoption and simplicity. Change management stays manageable because behavior shifts in small increments, staff retain their existing tools, and the architecture acquires few new dependencies, since the work remains within the core clouds and the trust layer the platform already provides. Retrieval-grounded assistance can leverage governed content to improve answer quality without altering model weights, and grounding model output in external, curated knowledge has become a standard technique for improving factual reliability in enterprise settings (Sharma, 2025). The observed limitations cluster around expressiveness. Multi-step work that crosses systems becomes hard to express through inline assistance, and teams that try to force such work into the pattern end up smuggling orchestration into prompts or into very large flows, where the control logic becomes opaque and brittle.

Approach B: The Autonomous Agent Tier

Approach B introduces a distinct orchestration tier. Typical instances include autonomous sales development agents that qualify and progress leads, service agents that resolve cases from intake to closure, and internal agents for information technology and human resources that call flows and APIs on a user's behalf. The unit of work is the end goal, and achieving the unit of work is a decision process in which the agent decides how to achieve the goal from a space of available actions. Like the shift from a single agent director to a multiple agent system, the literature has moved from a single planning model to multiple agents fulfilling an open-ended task through cooperative actions (Guo et al., 2024).

The conditions under which Approach B performs well also follow from its nature. It suits work that naturally spans multiple systems and several steps, where the relevant request is to resolve a billing issue end to end and not merely to draft a reply about it, and where the organization can treat the agent as a semi-autonomous digital worker that holds a clear scope and a defined set of tools, so that the agent absorbs the coordination that would otherwise fall on a person or on a tangle of inline logic.

The observed advantages cluster around the separation of concerns and oversight. Users continue to work in familiar interfaces while agents handle coordination behind the scenes, so the change to the user experience remains modest even as the underlying automation becomes substantial. The agent boundary provides a single, well-defined place to reason about the tools an agent may use, the guardrails that constrain it, and the monitoring that observes it, which supports the accountable oversight that responsible deployment requires (Chen et al., 2023). The observed limitations cluster around upfront effort and organizational commitment. Defining an agent's goals, tools, error handling, and handoffs to people demands considerable design work before any value appears, and the approach requires broader buy-in across the organization because it changes who performs the work.

Direct Comparison

Placing the two approaches side by side clarifies that they answer different questions. Approach A answers how to make an existing step faster or better, and Approach B answers who should carry out a multi-step task across systems. Before the criteria are discussed in prose, Table 1 sets them side by side so that the pattern of trade-offs can be read at a glance.

Table 1. Comparative summary of Approach A and Approach B across architectural and organizational criteria.

Criterion	Approach A: embedded assistance	Approach B: agent orchestration tier
Architectural position	Channel and business application layers	Dedicated AI orchestration layer
Typical scenarios	Reply suggestions, summaries, scoring, prompt-driven flows, inline grounded answers	Autonomous sales and service agents, internal agents calling flows and APIs
Best fit	Low-friction uplift inside adopted, single-team processes	Work that spans several systems and steps as a scoped digital worker
Unit of value	A faster or better bounded step	A reassigned multi-step task
Change management	Incremental and low-risk	Higher upfront design effort and broader buy-in
Architectural coupling	Few dependencies, stays in core clouds and trust layer	More dependencies, needs goals, tools, error handling, handoffs
Multi-step, cross-system work	Hard to express, orchestration smuggled into prompts or flows	Native strength, orchestration explicit at the agent boundary
Governance and monitoring	Diffuse, risk of interface clutter	Concentrated at a single, well-defined boundary
Suited adoption phase	First-generation adoption in a mature organization	Second-wave programs targeting real automation

The table exposes a consistent shape. Approach A wins on the dimensions that govern how easily a capability enters an organization, since it asks little of the existing process, introduces few dependencies, and stays inside the established trust surface, while Approach B wins on the dimensions that govern how much genuine automation a capability can express and how cleanly that automation can be governed, since it concentrates coordination and control at a single boundary that an architect can reason about. The dimension on which they most clearly diverge is the unit of value, because a faster step and a reassigned task are different kinds of outcomes with different consequences for the people whose work is affected. Figure 2 makes the control flow differences explicit by tracing how a request moves through each approach.

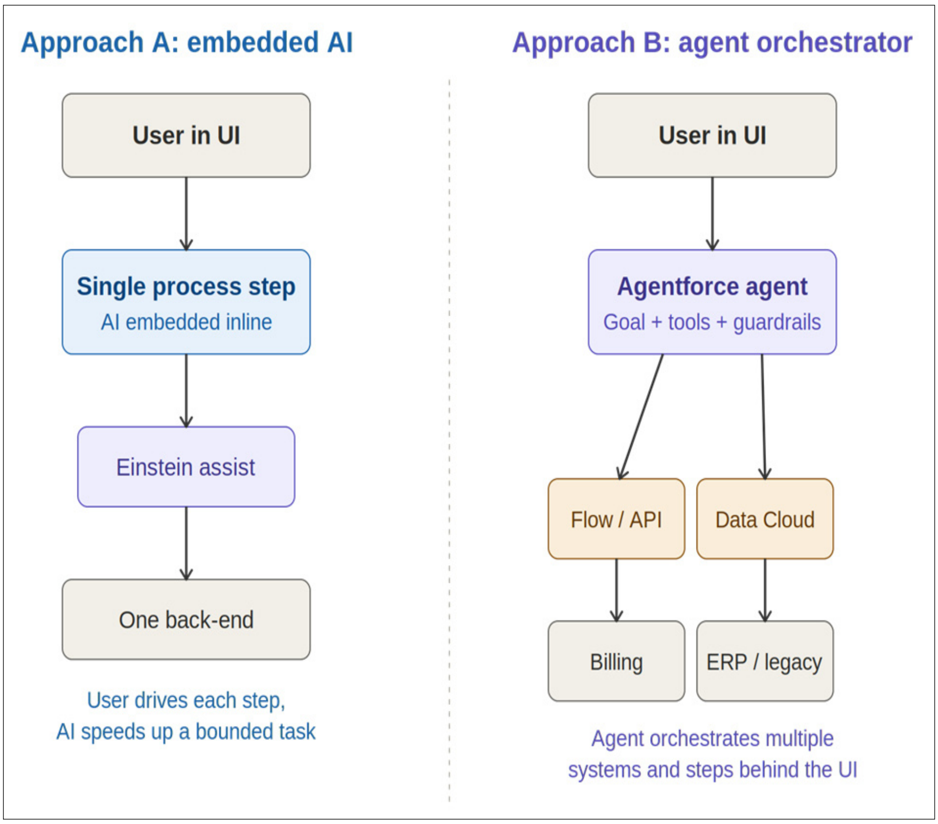

Fig. 2. Control flow under Approach A and Approach B, showing inline assistance against cross-system orchestration

In the left path the person drives each step, and the model accelerates one bounded task against a single back end, while in the right path the person states a goal once and the agent orchestrates calls to flows, data services, and several legacy systems before returning a result. The left path changes how fast the work goes while the right path changes who does the work, and this distinction determines which approach fits a given problem.

Supporting Architectural Dimensions

The choice between embedding and orchestration does not stand alone, because it sits inside a wider set of architectural commitments that practice has found to move together, where each of four neighboring decisions follows a logic in which a lightweight option serves early, tactical work while a more structured option serves strategic, cross-system work. The first decision concerns data access, where the contrast runs between direct calls into legacy application programming interfaces and a unified data layer such as a customer data platform. Direct calls win for isolated, tactical pilots in which one back end answers a narrow question. A unified data layer wins for anything strategic or cross-cloud, because it prevents the AI architecture from degenerating into a tangle of one-off integrations and gives every model a consistent governance surface on which the reliability of grounded output heavily depends (Sharma, 2025).

The second decision concerns integration style, where the contrast runs between point-to-point callouts and an integration platform or application programming interface gateway. Direct callouts work better for early pilots and for smaller organizations, since they bring something live quickly. A gateway or integration platform proves consistently safer and more maintainable for enterprises and for anything safety-critical, and the advantage grows once AI begins to generate a sizable volume of automated calls into fragile legacy systems, which parallels the wider challenge of implementing and scaling artificial intelligence across an organization (Haefner et al., 2023).

The third decision concerns the trust model, where the contrast runs between pervasive AI placed everywhere and scoped, high-value AI that keeps a person in the loop. Pervasive deployment tends to win the political argument during periods of enthusiasm, yet it frequently backfires on trust and governance once the breadth of automation outruns the controls around it. Scoped, high-value AI with human oversight wins in serious enterprises, and autonomy can expand from that base once metrics and guardrails are in place, since the case for keeping human judgment inside the loop rests on accountability and on the improvement in trust that follows when people retain meaningful oversight of automated decisions (Chen et al., 2023).

The fourth decision is how data migrations are handled, whether with a lift-and-shift approach or a transform-while-migrate approach. For AI-heavy roadmaps, teams that accept more difficulty during migration in order to clean and rationalize data tend to see better outcomes from later AI capability, while a two-step strategy that lifts and shifts first and then performs targeted cleanup before introducing AI can serve as a workable compromise where adoption risk runs high, since data readiness and governance frequently determine whether deployed capability delivers its promised value (Banh & Strobel, 2023). Read together, these four dimensions show that the central choice between Approach A and Approach B is embedded in a system of decisions, so the placement of AI signals the wider posture an organization has taken toward tactical speed or strategic durability.

DISCUSSION

The evidence assembled above points toward a conclusion that depends on organizational maturity and on the wave of adoption more than on any absolute ranking of the two approaches. For first-generation AI adoption inside a mature organization, embedded assistance tends to succeed more often, because it rides processes that staff have already accepted and asks little of the surrounding architecture, and the systematic study of AI in CRM has emphasized that successful integration depends on organizational and managerial fit at least as much as on technical capability (Ledro et al., 2023). For second-wave initiatives that target genuine automation, a dedicated agent tier yields better long-term outcomes than scattered inline features can, since the clean boundary of an orchestration tier becomes worth its higher upfront cost once an organization has learned where AI helps. This trajectory mirrors the research consensus that multi-agent orchestration extends what a single assistive model can accomplish, because coordination across systems is the capability that inline assistance struggles to express (Guo et al., 2024).

The reason the boundary falls where it does becomes clearer in light of this trajectory. The layered architecture places a natural seam between the model tier, where assistance augments a step, and the orchestration tier, where agents drive tasks. Crossing that seam by pushing orchestration into prompts and flows violates the separation that the architecture exists to maintain, and the violation shows up as opaque control logic and brittle behavior, while keeping orchestration in its own tier preserves the property that each capability has a defined place to live, which is the same property that makes the architecture governable.

The central choice connects outward to the supporting dimensions, reinforcing the maturity-based reading. A first-wave program that leans on embedded assistance pairs naturally with lighter data access, lighter integration, and lighter migration, and that combination is coherent for tactical work, while a second-wave program that leans on an agent tier pairs naturally with a unified data layer, a governed integration boundary, scoped autonomy under human oversight, and a more demanding migration posture. Trouble most often arises when an organization mixes the two postures, for instance, by standing up ambitious agents atop fragmented data and uncontrolled integration. A combined target architecture for large enterprises with heavy legacy favors scoped, high-value AI with a person in the loop during early phases, an agent-style orchestration tier for cross-system work, a unified data and grounding layer in place of direct calls into legacy, and an integration gateway as the main boundary to legacy systems for security and monitoring.

Several limits bound the analysis. The reasoning draws on patterns observed across enterprise deployments and on the research literature, and it does not rest on a controlled comparison of the two approaches within a single organization, which would be difficult to construct given the cost and duration of platform programs. The product capabilities that instantiate the two approaches continue to evolve quickly, so specific feature boundaries will shift even where the architectural seam holds. Future work could test the maturity-based reading through a longitudinal study of adoption programs.

CONCLUSION

The placement of artificial intelligence inside a customer relationship management platform is an architectural decision before it is a feature decision, and the two dominant patterns answer different questions. Embedding assistive models into existing interfaces and flows accelerates bounded steps that a person still drives and eases integration into an organization with little friction, making it a stronger choice for first-generation adoption in a mature setting. Introducing a separate orchestration tier of semi-autonomous agents reassigns multi-step, cross-system work to digital workers who coordinate behind familiar screens and provides a clean boundary for tools, guardrails, and monitoring, making it the stronger choice for second-wave programs that pursue real automation. The seven-layer reference architecture explains why the boundary between the two falls where it does, since the seam between the model tier and the orchestration tier marks the point at which augmenting a step gives way to driving a task, and the layered model gives architects a disciplined way to match the placement of capability to the maturity of the organization and the weight that AI carries in its strategy.

REFERENCES

1. Banh, L., & Strobel, G. (2023). Generative artificial intelligence. *Electronic Markets, 33*, Article 63. https://doi.org/10.1007/s12525-023-00680-1
2. Chen, X., Wang, X., & Qu, Y. (2023). Constructing ethical AI based on the “human-in-the-loop” system. *Systems, 11*(11), Article 548. https://doi.org/10.3390/systems11110548
3. Ettinger, A. (2025). *Enterprise architecture as a dynamic capability for scalable and sustainable generative AI adoption: Bridging innovation and governance in large organisations*. arXiv. https://doi.org/10.48550/arXiv.2505.06326
4. Guo, T., Chen, X., Wang, Y., Chang, R., Pei, S., Chawla, N. V., Wiest, O., & Zhang, X. (2024). Large language model based multi-agents: A survey of progress and challenges. In K. Larson (Ed.), *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24* (pp. 8048–8057). International Joint Conferences on Artificial Intelligence Organization. https://doi.org/10.24963/ijcai.2024/890
5. Haefner, N., Parida, V., Gassmann, O., & Wincent, J. (2023). Implementing and scaling artificial intelligence: A review, framework, and research agenda. *Technological Forecasting and Social Change, 197*, Article 122878. https://doi.org/10.1016/j.techfore.2023.122878
6. Ledro, C., Nosella, A., & Vinelli, A. (2022). Artificial intelligence in customer relationship management: Literature review and future research directions. *Journal of Business & Industrial Marketing, 37*(13), 48–63. https://doi.org/10.1108/JBIM-07-2021-0332
7. Ledro, C., Nosella, A., & Dalla Pozza, I. (2023). Integration of AI in CRM: Challenges and guidelines. *Journal of Open Innovation: Technology, Market, and Complexity, 9*(4), Article 100151. https://doi.org/10.1016/j.joitmc.2023.100151
8. Nicolescu, L., & Tudorache, M. T. (2022). Human-computer interaction in customer service: The experience with AI chatbots—A systematic literature review. *Electronics, 11*(10), Article 1579. https://doi.org/10.3390/electronics11101579
9. Sharma, C. (2025). *Retrieval-augmented generation: A comprehensive survey of architectures, enhancements, and robustness frontiers*. arXiv. https://doi.org/10.48550/arXiv.2506.00054