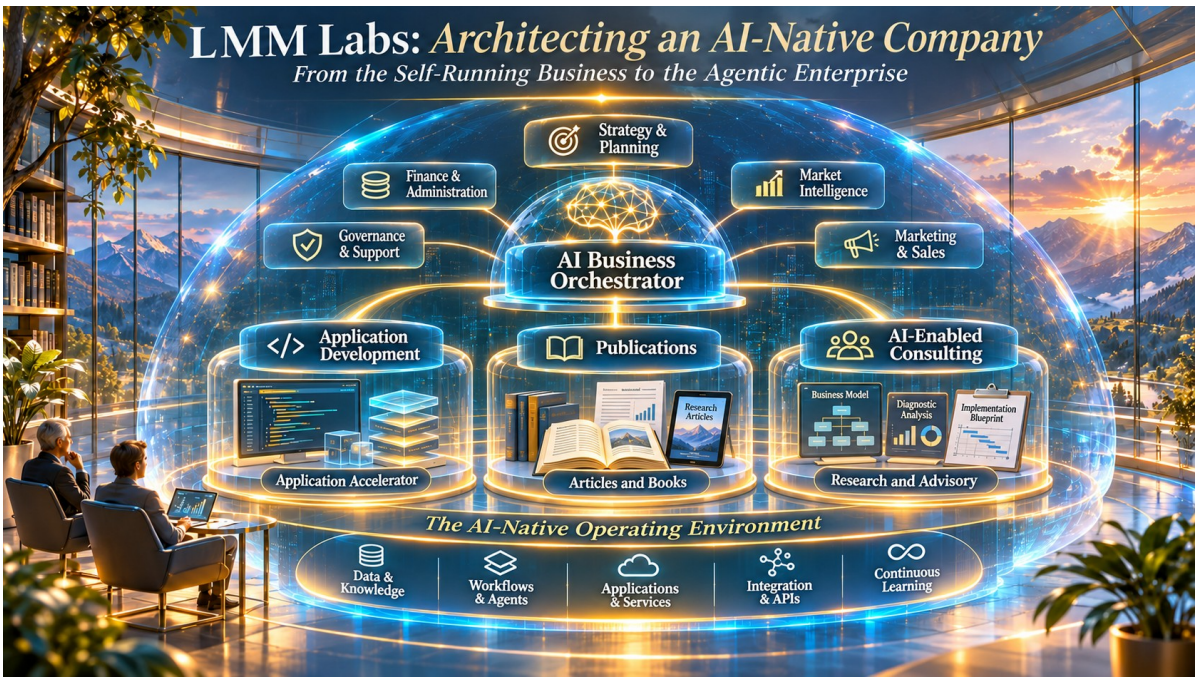


Opening illustration. An artistic representation of LMM Labs' proposed AI-native company: application development, publications and consulting within a shared operating environment. AI-generated artwork.

LMM Labs: Architecting an AI-Native Company

Organizational foundations, system architecture, and a prospective design study

Lee J. Farretta and Matthew Farretta | LMM Labs
Prepared in collaboration with ChatGPT

Abstract

This article proposes an architecture for running a company's core business functions through AI agents under human direction. It draws on research in organizational coordination, delegation and autonomic computing to define how those agents would work together (Baird & Maruping, 2021; Kephart & Chess, 2003; Malone & Crowston, 1994). Human leadership would set objectives and define the agents' authority. A Business Orchestrator would plan work, coordinate specialist agents and assess results. Shared organizational records would retain decisions and commitments across tasks, while programmatic controls would enforce permissions and route exceptions to human decision makers. LMM Labs is the proposed setting for constructing and evaluating this architecture. Its Application Accelerator would provide engineering capabilities, while the Business Orchestrator would coordinate the company's operations. Application development, publications and AI-enabled consulting would draw on the same core business functions. Evaluation would examine outcome quality, continuity, human oversight, operating efficiency and adaptability. The central question is whether this structure can support dependable operation with limited routine human intervention as the mix of products and services changes.

1. Introduction

A company must coordinate work across functions, allocate resources and make decisions with incomplete information. Organizational research explains why these activities require shared information and clear relationships among responsibilities (Galbraith, 1974; Malone & Crowston, 1994). In the AI-native company examined here, software agents would perform much of that work under authority delegated by human leadership. Baird and Maruping's (2021) framework for delegation to agentic systems provides the basis for defining those responsibilities. This article develops an architecture for coordinating them and uses LMM Labs as a proposed case study. Design-science research provides the method: build the system and evaluate how well it serves its intended purpose (Hevner et al., 2004).

2. Historical and intellectual foundations

Delegating work to AI does not remove the coordination problems that organizations already face. Galbraith (1974) linked uncertainty to an organization's need to process information. Malone and Crowston (1994) examined coordination as the management of dependencies among activities, including work that shares resources or depends on another activity's results. Applied to an AI-native company, these ideas suggest that agents need a common account of the work they are pursuing and the commitments that constrain it. The architecture developed here therefore treats coordination as an explicit responsibility: it must connect each agent's work to the objectives, resources and activities on which that work depends.

Business process research explains why those relationships should be designed alongside the technology. Davenport and Short (1990) treated information technology and process redesign as interdependent, with particular attention to processes that cross functional boundaries. Brynjolfsson and Hitt (2000) likewise emphasized the organizational changes that complement investment in information technology. For the architecture developed here, the implication is to define how agents contribute to complete business processes. Each responsibility must specify the dependencies and handoffs through which work moves between functions.

Autonomic computing supplied a technical precedent for managing such work through feedback. Kephart and Chess (2003) described systems that would configure, optimize, repair and protect themselves under high-level policies. An autonomic manager would monitor its environment, analyze information, plan actions and execute them, while coordinating with other elements. Their vision also encompassed businesses and automated enterprises. The LMM Labs architecture applies this feedback principle to delegated business responsibilities: agents act within stated limits, assess results and adjust their work as conditions change.

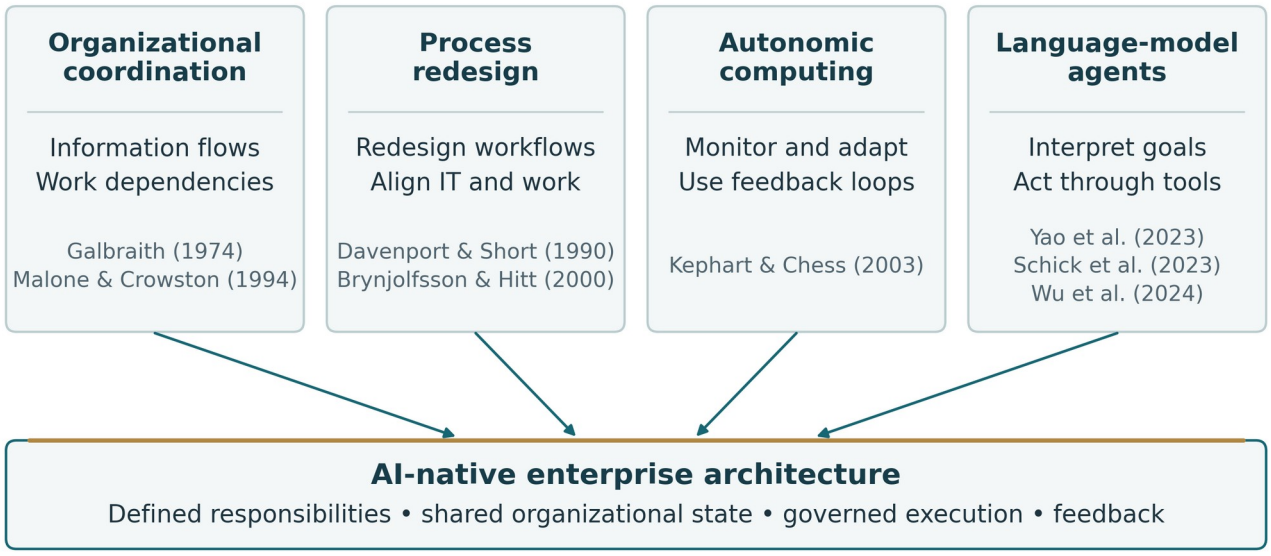
LMM Labs' earlier *The Self-Running Business (SRB): An Introduction to AI-Driven Business Autonomy* supplies the immediate foundation for this design. Chapter 9 describes software-defined business functions and an orchestration layer that coordinates them within a common business architecture. Chapter 10 extends that approach to AI-enabled coordination across the value chain (LMM Labs, n.d., Chapters 9–10). The present article develops those ideas through contemporary agent research, with

particular attention to delegated authority, shared organizational records and evaluation. The aim is to explain how these elements would work together in a company.

Delegation research helps specify the relationship between human leadership and software agents. Baird and Maruping (2021) identify appraisal, distribution and coordination of responsibilities as parts of delegation between people and agentic information systems. Li et al. (2026) distinguish the capabilities of an individual agent from the coordination and supporting capabilities of an enterprise agent system. Together, these perspectives suggest that an agent's task must be defined alongside its authority and its relationship to other participants. Figure 1 brings these research foundations together in the architecture developed in this article.

Figure 1

Research Foundations of the Proposed Enterprise Architecture



Note. The source attributions within the figure identify the research foundations. The arrows and the resulting architecture represent the authors' synthesis.

Current and forthcoming business-school courses address these organizational questions. Harvard Business School Online's *Designing Organizations for Agentic AI* covers organizational redesign, value creation and control in an online certificate course (Harvard Business School Online, n.d.). Stanford Graduate School of Business's MBA elective *The AI-Powered Org: Evolution, Rebirth, or Death?* examines AI's organizational consequences through established theory (McDowell, 2026). Kellogg's catalog lists *Building and Managing an AI Native Company* for Spring 2027 (Northwestern University, Kellogg School of Management, n.d.).

3. Why the architecture is feasible now

Recent agent research provides mechanisms for putting these organizational ideas into practice. Li et al. (2026) describe agentic systems in terms of goals, environmental interaction, tools, memory, reasoning and actions. The LMM Labs design would connect these capabilities to application interfaces, data services and workflows. This extends the software-defined business functions described in *The Self-*

Running Business (LMM Labs, n.d., Chapter 9). Three areas of research are particularly relevant: retrieving information, acting through tools and coordinating multiple agents.

Retrieval-augmented generation (RAG) connects a language model to information retrieved from external sources. Lewis et al. (2020) demonstrated this approach on knowledge-intensive language tasks. In the LMM Labs architecture, retrieval would give agents access to maintained organizational knowledge. Shared business records would serve a different purpose: recording what the company has authorized, committed to and completed. The architecture keeps these records distinct from background documents so that agents can coordinate their actions around current commitments. This design choice applies Malone and Crowston's (1994) account of dependencies among activities and shared resources.

ReAct demonstrated an interaction pattern in which reasoning and actions are interleaved with observations from an environment (Yao et al., 2023). Toolformer showed that models could learn when and how to invoke selected tools within its experimental setting (Schick et al., 2023). AutoGen demonstrated how agents, tools, human participation and programmed control could be combined to perform tasks (Wu et al., 2024). These studies provide the technical basis for agents that consult information, act through software interfaces and coordinate with other participants. The LMM Labs architecture would use those mechanisms to carry out delegated work across business functions.

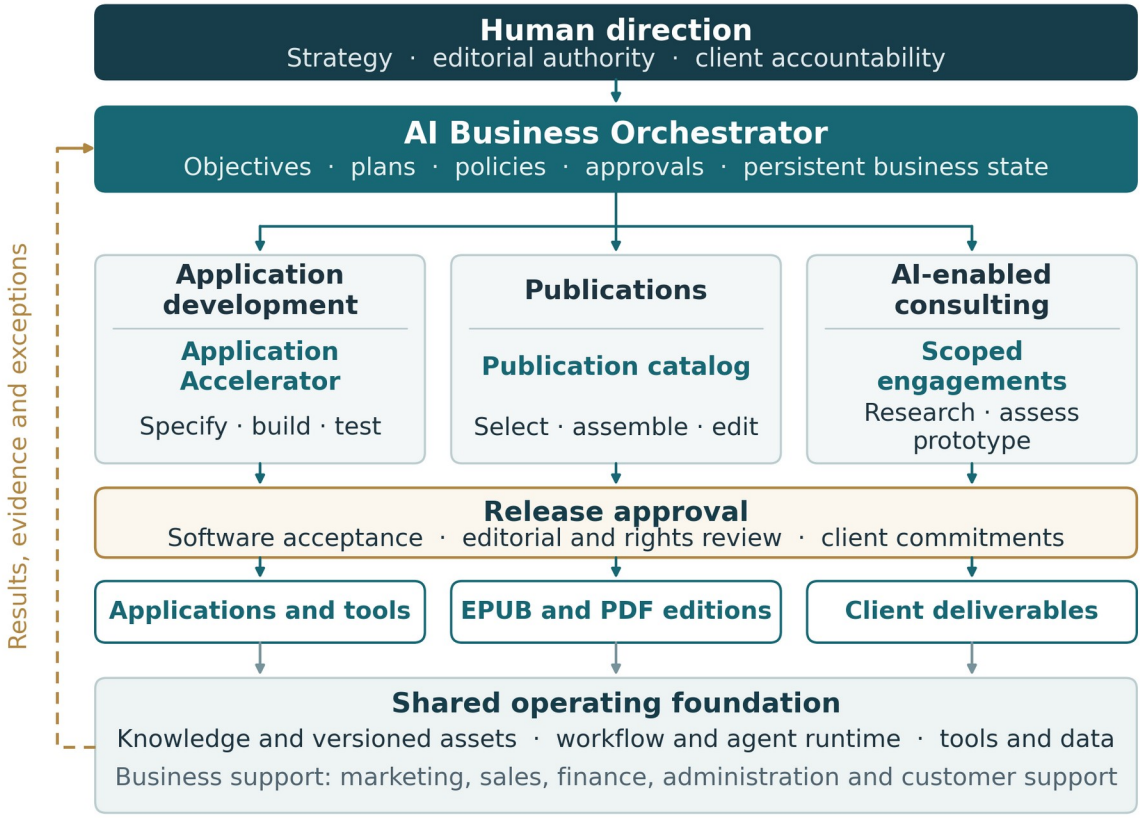
Enterprise benchmarks provide ways to assess these capabilities on business tasks. TheAgentCompany evaluated 175 tasks in a simulated software company; the strongest baseline reported in its 2025 study fully completed 30.3% of them (Xu et al., 2025). EnterpriseBench combines static knowledge tasks with interactive enterprise decisions and reports differences in performance across methods and models (Yang et al., 2026). These studies evaluate tasks and simulations, so their results do not establish reliable operation of an entire company over time. For LMM Labs, the next step would be to integrate the demonstrated mechanisms and evaluate their performance across the company's functions.

4. Architecture of an AI-native company

The architecture connects human direction, orchestration, specialist work, shared records and assurance. Figure 2 shows how these responsibilities connect. Human direction guides the Business Orchestrator, which coordinates work across shared business capabilities and the three product lines. The design combines dependency management from coordination theory with the composition of agents and tools demonstrated in AutoGen (Malone & Crowston, 1994; Wu et al., 2024). Authority is controlled separately from task assignment, applying the security principles discussed by Saltzer and Schroeder (1975). The following subsections explain each part of that arrangement.

Figure 2

LMM Labs Business Orchestration and Product Architecture



Note. The design separates human direction, orchestration and production, drawing on delegation and agent composition (Baird & Maruping, 2021; Wu et al., 2024). Release approval is a distinct authorization control (Saltzer & Schroeder, 1975). Shared business functions support all three product lines.

4.1 Organizational intent and decision rights

Human leadership sets objectives, priorities and constraints, then defines which decisions agents may make. These decision rights establish the scope of delegated work and the circumstances that require human judgment. Applying Baird and Maruping's (2021) delegation framework, each responsibility would be specified together with its authority. A human request could initiate or redirect work, while standing policies would determine which actions agents may take. Recurring work could then proceed within those policies without a new instruction for every action.

4.2 Orchestration and specialist capabilities

The Business Orchestrator turns organizational objectives into coordinated assignments. It identifies dependencies, assigns work to specialist agents, assesses their results and revises plans when conditions change. AutoGen demonstrates how agent roles and control arrangements can be composed, while Li et al. distinguish coordination across an enterprise system from the capabilities of its individual agents (Li et al., 2026; Wu et al., 2024). The orchestrator is therefore defined here by its coordination responsibilities. It could use multiple cooperating components, provided they maintain a shared account of objectives, commitments and unresolved decisions.

4.3 Shared organizational state

That shared account is the company's organizational state: its authorized objectives, assigned responsibilities, resource commitments, accepted results and supporting decision records. This record would persist beyond individual agent sessions so that subsequent work can take account of prior decisions and obligations. The design combines Malone and Crowston's (1994) treatment of dependencies with Li et al.'s (2026) account of memory in agent systems. Its purpose is to keep agents working from consistent commitments as tasks change hands or models are replaced. Operating trials would test whether the records actually preserve that continuity.

4.4 Execution and authority boundaries

Execution combines agent judgment with programmatic control. AutoGen demonstrates how agents, tools and human participation can operate within programmed interactions (Wu et al., 2024). In LMM Labs, prescribed workflows would govern stable sequences, while agents would plan less structured work. Both would operate within enforced authority boundaries. Saltzer and Schroeder's (1975) principles of least privilege and complete mediation provide the rationale: components receive only the authority needed for their tasks, and access is checked against the applicable authorization. Execution controls would therefore check an agent's proposed action before allowing it to affect company resources or commitments.

4.5 Feedback, assurance and human judgment

Feedback connects completed work to the next decision. The system would compare results with intended outcomes, identify unresolved conditions and return that information to the orchestrator. This applies the feedback principle of autonomic computing and NIST's emphasis on evaluation and monitoring (Kephart & Chess, 2003; Tabassi, 2023). The orchestrator could adjust work within its delegated authority; changes to objectives, contested judgments and exceptions outside that authority would go to human leadership. Operating evidence would also help leadership decide whether to revise the policies governing future work.

5. Core business functions as coordinated capabilities

The company's core functions support every product line. Finance, operations, governance and customer support remain business responsibilities whether the company produces software, publications or consulting services. Table 1 assigns these responsibilities within the LMM Labs design. It applies the software-defined functions described in *The Self-Running Business*, together with research on cross-functional processes and dependencies (Davenport & Short, 1990; LMM Labs, n.d., Chapter 9; Malone & Crowston, 1994). The eight categories specify the work the company must coordinate; they do not require a separate software agent for each department.

Table 1

Core Business Capabilities and Proposed Responsibilities

Core capability	Architectural responsibility
Strategy and planning	Align priorities and resource commitments with the company's objectives.
Market intelligence	Assess external information and identify uncertainty that affects business decisions.
Marketing and sales	Align customer demand and external commitments with the company's capacity and approved offerings.
Finance and accounting	Maintain reliable records of resources, obligations and financial constraints.
Operations, IT and administration	Maintain the services, information flows and continuity needed by the other functions.
Customer relationships and support	Maintain customer commitments, relationship history and support responsibilities.
People, capabilities and knowledge	Manage the expertise, reusable knowledge and responsibilities of human and software participants.
Governance and assurance	Maintain approved decision rights and standards, and assess whether actions and results comply.

Note. The eight categories define responsibilities for the LMM Labs case. Their conceptual basis is software-defined business functions, cross-functional process design and coordination of dependencies (Davenport & Short, 1990; LMM Labs, n.d., Chapter 9; Malone & Crowston, 1994).

These functions depend on one another. Organizational priorities guide resource allocation, and external commitments create obligations for production and support. The orchestrator must account for those relationships when assigning work and accepting results. Coordination theory explains the underlying dependencies, while Cemri et al. document failures in inter-agent alignment and task verification (Cemri et al., 2025; Malone & Crowston, 1994). Figure 3 applies these findings by placing specialist agents within a common operating foundation. A completed task would be assessed both against its own requirements and against the commitments it creates or changes elsewhere in the business.

Figure 3

Orchestration of Core Business Functions



Note. Specialist agents perform the core business functions under the orchestrator's direction. The design applies dependency management, feedback and defined human–AI roles (Kephart & Chess, 2003; Malone & Crowston, 1994; Tabassi, 2023). These functions support the product lines shown in Figure 2.

The way a function is performed can change without changing its purpose. A responsibility may be assigned to a deterministic software service, one agent, several cooperating agents or a human specialist. Its inputs, expected outcomes and authority would remain explicit. This design choice combines the delegation framework with the flexible composition of agent systems (Baird & Maruping, 2021; Wu et al., 2024). It is intended to let LMM Labs change the technology performing a function while preserving the company's responsibilities and commitments.

6. LMM Labs as a prospective case study

LMM Labs would test whether the same business architecture can support application development, publications and AI-enabled consulting. These product lines require different production processes, but the design assigns them common business functions. The case extends the software-defined functions and orchestration model described in *The Self-Running Business* into an architecture that can be built and evaluated (LMM Labs, n.d., Chapters 9–10). Its central hypothesis is that shared functions can remain effective as the company changes its products and services.

Two complementary platforms would support the company. The Application Accelerator would create and maintain software. The Business Orchestrator would coordinate the company's operations, including work that uses the accelerator. Business requirements would guide engineering, and engineering would produce capabilities that the business could adopt. Creating a capability would not, however, authorize its deployment or use. That distinction applies the delegation and security principles developed above: software production and permission to act on behalf of the company are separate responsibilities (Baird & Maruping, 2021; Saltzer & Schroeder, 1975).

The three product lines would exercise the shared architecture in different ways. Application development would produce software and support its continued use. Publications would turn research and articles into edited publications and book collections. AI-enabled consulting would apply shared expertise within defined professional engagements. Raji et al. (2020) emphasize documentation and evidence throughout an AI system's lifecycle. Applying that principle to this case, each product line would retain records identifying its sources, the criteria used to accept its work and who is responsible for the resulting deliverables.

The publishing design separates content, editorial authority and distribution. Selected existing and new material could be assembled into collections for Kindle Direct Publishing or a direct-download store. The design objective is to retain common knowledge and editorial responsibilities as distribution channels change. This applies research connecting technology with work processes and organizational practices to a specific product line (Brynjolfsson & Hitt, 2000; Davenport & Short, 1990). Evaluation would examine whether those shared responsibilities remain effective when the publishing process changes, and whether the same principle holds across the other product lines.

The software architecture also needs an organizational foundation. As LMM Labs is established, ownership and responsibility for assets, obligations and decisions would need to be assigned. Those assignments would determine who may delegate work, set limits and approve changes. NIST's treatment of responsibility and human–AI roles supports making these allocations explicit (Tabassi, 2023). In this design, organizational responsibilities would provide the basis for the permissions enforced by the software.

Development would begin with the engineering environment and initial Application Accelerator, followed by parallel construction of the accelerator and Business Orchestrator. Their interfaces would specify the capabilities available, the authority required to use them and the criteria for accepting results. The study would examine whether new capabilities can be added while the company maintains consistent records of resources, commitments and responsibilities. This is a design-science question about the usefulness of a constructed system: can it support different product configurations with dependable results and a manageable demand for human oversight (Hevner et al., 2004)?

7. Authority, accountability and assurance

The scope of delegation would depend on the work and the evidence of its performance. NIST's AI Risk Management Framework recognizes different human–AI roles and configurations of human involvement (Tabassi, 2023). Applying that approach, LMM Labs would consider the consequences of error, the reversibility of actions and the quality of available evidence when assigning authority. Work with defined

limits and demonstrated performance could proceed under standing authority. Specified decisions and unresolved exceptions would remain with human leadership.

Oversight requires evidence about what the system actually does. Raji et al. (2020) emphasize documentation and examination throughout development and deployment. For LMM Labs, the records would connect assigned objectives, authority used, evidence considered and accepted results. Evaluation would compare actual outcomes with criteria established independently of the acting agent's account of its success. This applies lifecycle auditing and NIST's emphasis on documented measurement to the assessment of delegated work (Raji et al., 2020; Tabassi, 2023). Agent messages could contribute to that record, but they would not substitute for evidence of outcomes.

Adding agents can increase the effort required to coordinate and verify work. Cemri et al. (2025) identify system-design, inter-agent alignment and task-verification failures in studied multi-agent systems. Sculley et al. (2015) describe how machine-learning systems accumulate dependencies, feedback effects and maintenance burdens beyond the model itself. These findings support evaluating the additional interactions introduced by each agent role. LMM Labs would add roles where they provide a demonstrable capability or separation of responsibility, while keeping their interactions observable and providing for controlled recovery from failure.

The Business Orchestrator would be subject to the same discipline as the agents it coordinates. Changes to its objectives, operating rules or access privileges could affect work throughout the company. Each change would therefore be evaluated against approved purposes, delegated authority and operating evidence. Least privilege limits the reach of a defective or misdirected component, while lifecycle auditing provides a basis for reassessing the system as it changes (Raji et al., 2020; Saltzer & Schroeder, 1975). The design applies those principles to changes in the orchestrator itself.

8. Building and evaluating the company

Construction and evaluation should proceed together. Design-science research treats a system's usefulness as something to be demonstrated through evaluation (Hevner et al., 2004). For LMM Labs, the first stage would establish organizational responsibilities and shared records. The next would introduce agents with bounded responsibilities, followed by integration across functions and product lines. Each stage would assess whether work remains coordinated as the system's scope increases. Broader delegation would follow evidence that the preceding stage operates dependably.

Evaluation would examine individual capabilities, their integration and sustained operation. Component tests would assess capabilities against stated requirements. Integrated tests would examine dependencies among functions and the consistency of shared records. Operating trials would assess the company's ability to continue through changing priorities, incomplete information and service interruptions. This progression applies design-science evaluation to the architecture as a whole (Hevner et al., 2004). TheAgentCompany and EnterpriseBench offer precedents for task-based assessment and interactive decision tests (Xu et al., 2025; Yang et al., 2026). Table 2 sets out the questions the LMM Labs study would address.

Table 2

Dimensions for Evaluating the LMM Labs Architecture

Dimension	Organizational question	Evidence sought
Outcome quality	Does delegated work satisfy the organization's requirements?	Results evaluated against criteria defined before execution.
Coordination and continuity	Does work remain coordinated across functions and over time?	Consistent shared records, fulfilled commitments and recovery after interruptions.
Human oversight	How much supervision does dependable operation require?	Human effort, consequential interventions and unresolved exceptions relative to completed work.
Operating efficiency	Does orchestration improve efficiency after coordination and assurance are included?	Total operating effort and resource consumption for comparable outcomes.
Adaptability	Can the operating structure accommodate a changed product mix?	Reuse of shared functions, with authority reassessed as production requirements change.

Note. These study criteria draw on design-science evaluation, task and simulation benchmarks, and monitoring during operation (Hevner et al., 2004; Sculley et al., 2015; Tabassi, 2023; Xu et al., 2025; Yang et al., 2026). They specify evidence to collect during the LMM Labs study.

The study would compare prescribed workflows, single-agent arrangements and multi-agent arrangements under comparable requirements where the work permits. It would assess outcome quality and continuity alongside the effort spent on coordination, verification and human oversight. Cemri et al.'s (2025) analysis of multi-agent failures motivates examining the costs and failure modes that coordination can introduce. Combined with design-science evaluation, these comparisons would help determine which arrangements justify broader delegated responsibility in LMM Labs (Hevner et al., 2004).

Sustained operation must be evaluated over time. Research on technical debt and AI risk management calls attention to dependencies that change as systems operate and evolve (Sculley et al., 2015; Tabassi, 2023). The LMM Labs study would therefore record completed outcomes together with the human and technical effort needed to maintain them. It would also examine whether responsibilities, shared records and authority remain consistent as data and software change. Changes in the balance among application development, publications and consulting would test whether the same operating structure can support a different product mix.

9. Architectural implications and limits

The central architectural choice is to define each business capability by its responsibility, information needs, authority and expected outcomes. The software performing that responsibility could then change while its organizational purpose remains explicit. This choice follows from combining research on organizational information processing, dependencies and delegation (Baird & Maruping, 2021; Galbraith, 1974; Malone & Crowston, 1994). For LMM Labs, the intended benefit is continuity as agents,

models or services are replaced. Evaluation would need to establish whether that separation actually preserves commitments and coordinated work.

The reviewed studies demonstrate mechanisms for agent action and ways to evaluate performance on tasks and simulations (Xu et al., 2025; Yang et al., 2026; Yao et al., 2023). They do not establish how a complete company would perform when responsibilities persist and the results of one function affect another over time. The LMM Labs study would address that question in its own operating context. Its findings would initially apply to the products, responsibilities and conditions evaluated. Applying the architecture to organizations with substantially different work or obligations would require further evaluation.

10. Conclusion

The architecture developed here connects delegated work to organizational objectives, shared commitments and human authority. Coordination and delegation research explain why the responsibilities assigned to agents must be considered together with their dependencies and decision rights (Baird & Maruping, 2021; Malone & Crowston, 1994). Research on agent systems supplies mechanisms for coordinating executable work, while security and lifecycle assurance provide principles for controlling and evaluating it (Raji et al., 2020; Saltzer & Schroeder, 1975; Wu et al., 2024). LMM Labs would bring these elements together through the parallel development of its Application Accelerator and Business Orchestrator. Application development, publications and consulting would provide different settings in which to evaluate the shared business functions. The central test is whether those functions can operate dependably, with limited routine human intervention, as the company’s products and services change.

References

Baird, A., & Maruping, L. M. (2021). The next generation of research on IS use: A theoretical framework of delegation to and from agentic IS artifacts. *MIS Quarterly*, *45*(1), 315–341. https://doi.org/10.25300/MISQ/2021/15882

Brynjolfsson, E., & Hitt, L. M. (2000). Beyond computation: Information technology, organizational transformation and business performance. *Journal of Economic Perspectives*, *14*(4), 23–48. https://doi.org/10.1257/jep.14.4.23

Cemri, M., Pan, M. Z., Yang, S., Agrawal, L. A., Chopra, B., Tiwari, R., Keutzer, K., Parameswaran, A., Klein, D., Ramchandran, K., Zaharia, M., Gonzalez, J. E., & Stoica, I. (2025). Why do multi-agent LLM systems fail? In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, & N. Chen (Eds.), *Advances in neural information processing systems* (Vol. 38). Curran Associates, Inc. https://doi.org/10.52202/085713-4082

Davenport, T. H., & Short, J. E. (1990). The new industrial engineering: Information technology and business process redesign. *Sloan Management Review*, *31*(4), 11–27. https://dspace.mit.edu/handle/1721.1/48613

Galbraith, J. R. (1974). Organization design: An information processing view. *Interfaces*, *4*(3), 28–36. https://doi.org/10.1287/inte.4.3.28

Harvard Business School Online. (n.d.). *Designing organizations for agentic AI* [Online course description and syllabus]. Retrieved November 26, 2026, from https://online.hbs.edu/courses/designing-organizations-agentic-ai

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, *28*(1), 75–105. https://doi.org/10.2307/25148625

Kephart, J. O., & Chess, D. M. (2003). The vision of autonomic computing. *Computer*, *36*(1), 41–50. https://doi.org/10.1109/MC.2003.1160055

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (Vol. 33, pp. 9459–9474). Curran Associates, Inc. https://proceedings.nips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html

Li, M. M., Reinhard, P., & Leimeister, J. M. (2026). Agentic AI: From autonomous agents to enterprise agentic AI systems. *Business & Information Systems Engineering*, *68*(5), 1191–1204. https://doi.org/10.1007/s12599-026-01009-w

LMM Labs. (n.d.). *The self-running business (SRB): An introduction to AI-driven business autonomy* [Manuscript]. https://lmmlabs.com/research/

Malone, T. W., & Crowston, K. (1994). The interdisciplinary study of coordination. *ACM Computing Surveys*, *26*(1), 87–119. https://doi.org/10.1145/174666.174668

McDowell, M. (2026, September 24). *Back to class: The AI-powered org*. Stanford Graduate School of Business. https://www.gsb.stanford.edu/experience/news-history/back-class-ai-powered-org

Northwestern University, Kellogg School of Management. (n.d.). *Building and managing an AI native company* [Course catalog entry, AIML-938-0]. Retrieved November 26, 2026, from https://www6.kellogg.northwestern.edu/CourseCatalog/coursecatalog/coursedetail?coursecatalogid=208121

Raji, I. D., Smart, A., White, R. N., Mitchell, M., Gebru, T., Hutchinson, B., Smith-Loud, J., Theron, D., & Barnes, P. (2020). Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (pp. 33–44). Association for Computing Machinery. https://doi.org/10.1145/3351095.3372873

Saltzer, J. H., & Schroeder, M. D. (1975). The protection of information in computer systems. *Proceedings of the IEEE*, *63*(9), 1278–1308. https://doi.org/10.1109/PROC.1975.9939

Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., & Scialom, T. (2023). Toolformer: Language models can teach themselves to use tools. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, & S. Levine (Eds.), *Advances in neural information processing systems* (Vol. 36, pp. 68539–68551). Curran Associates, Inc. https://doi.org/10.52202/075280-2997

Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., & Dennison, D. (2015). Hidden technical debt in machine learning systems. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 28). Curran Associates, Inc. https://proceedings.neurips.cc/paper/2015/hash/86df7dcfd896fcaf2674f757a2463eba-Abstract.html

Tabassi, E. (2023). *Artificial intelligence risk management framework (AI RMF 1.0)* (NIST AI 100-1). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.AI.100-1

Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., Jiang, L., Zhang, X., Zhang, S., Liu, J., Awadallah, A., White, R. W., Burger, D., & Wang, C. (2024). AutoGen: Enabling next-gen LLM applications via multi-agent conversations. In *First Conference on Language Modeling*. https://www.microsoft.com/en-us/research/wp-content/uploads/2023/08/LLM_agent.pdf

Xu, F. F., Song, Y., Li, B., Tang, Y., Jain, K., Bao, M., Wang, Z. Z., Zhou, X., Guo, Z., Cao, M., Yang, M., Lu, H. Y., Martin, A., Su, Z., Maben, L. M., Mehta, R., Chi, W., Jang, L., Xie, Y., … Neubig, G. (2025). TheAgentCompany: Benchmarking LLM agents on consequential real world tasks. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, & N. Chen (Eds.), *Advances in neural information processing systems* (Vol. 38). Curran Associates, Inc. https://doi.org/10.52202/085713-0315

Yang, M., Pan, Y., Piao, J., Song, D., Gong, Y., & Li, Y. (2026). EnterpriseBench: Benchmarking LLM agents on enterprise-level strategic reasoning and decision-making. In H.-Y. Lee, S. Dev, J. Li, & D. Elliott (Eds.), *Findings of the Association for Computational Linguistics: EMNLP 2026* (pp. 19659–19682). Association for Computational Linguistics. https://doi.org/10.18653/2026.findings-emnlp.932

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). ReAct: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=WE_vluYUL-X

Appendix

Glossary

The following definitions specify how terms are used in the proposed architecture. Citations identify their conceptual foundations.

Agent. A software component that interprets objectives and selects actions using a model, tools and observations within a defined scope (Li et al., 2026).

AI-native company. In this article, a company designed to delegate much of the work of its core business functions to software agents under human authority and programmatic control. This definition applies the delegation framework to organizational design (Baird & Maruping, 2021).

Assurance. The collection and assessment of evidence that a system meets its stated requirements, including examination throughout its development and operation (Raji et al., 2020; Tabassi, 2023).

Business Orchestrator. The part of the architecture that plans and coordinates delegated work across core business functions, using shared records and feedback (LMM Labs, n.d., Chapter 9; Malone & Crowston, 1994).

Decision rights. The authority assigned to a person or system to make specified decisions. In this design, that authority also sets the limits on further delegation (Baird & Maruping, 2021; Tabassi, 2023).

Organizational state. The maintained record of authorized objectives, assigned responsibilities, resource commitments and accepted results. In this design, it supports coordination across activities and agent sessions (Li et al., 2026; Malone & Crowston, 1994).