

Multi-Agent Orchestration for Agentic Commerce: Coordinating Shopping, Payment, Rewards, and Fraud Agents in Real-Time Financial Transactions

Muralidharan Lakshmanan ^{1,*}

¹ Independent Researcher, Atlanta, Georgia, United States

Email: ¹ muralidharan.lakshmanan@ieee.org

*Corresponding Author

Abstract—Agentic commerce is shifting the process of digital commerce buying away from a set of stand-alone recommenders and towards orchestrated autonomous services that can identify items, approve payment, earn rewards and manage transaction risk all in one real-time interaction. This study suggests a multi-agent orchestration system where shopping, payment, rewards and fraud agents are linked together via a central coordinator and their functional boundaries are kept clear. The study created a synthetic data set of 127 transaction cases with varying customer use cases, merchant contexts, payment terms, reward opportunities and fraud review signals for the evaluation. The dataset has been designed for experimentation and re-experimentation and does not include any personal information from other sources. Data preparation of transaction records, coordination of scenario execution and organization of experimental outputs and result graphs were done using Python, pandas were used for structured data handling and Matplotlib for the result graphs. The decision flow continuity, reliability of authorization, alignment of incentives, fraud control, consistency of response and readiness of audit are the key assessment points. Results show that by having a coordinated agent that takes care of managing the order of messages, the shared context, the policy checks and exception handling, it is more effective than letting independent services operate without being aware of each other across transactions. The architecture is suitable for financial commerce platforms that need quick decisions and secure automated responses, improved accountability, and customer experience throughout the entire purchasing process.

Keywords—Agentic commerce; multi-agent orchestration; Payment coordination; Rewards automation; Fraud control.

I. INTRODUCTION

Today, digital commerce is at a stage where software agents can be involved in activities that in the past were taken care of by activities carried out by different users and back-end services as described in the previous work [8]. A customer can talk about a purchase intention, be shown what the security controls have determined is a more appropriate product, approve a payment option that is more convenient to him, withdraw a reward that he is eligible for, and make a purchase, all in the natural language he is using. It relies on the ability of a number of specialized agents to cooperate

without loss of context, duplicate effort or make contradictory decisions as shown in previous studies [3]. Real-time financial transactions like payments are especially sensitive because if the message doesn't get through or if there's a disparity in how the policy is interpreted or if there's an uncoordinated retry, it could impact the trust in the service, merchant acceptance, payment completion and security review. Thus, multi-agent orchestration is a layer of operation, which can bind autonomous actions to a series of controlled transactions, as studied in previous academic research [11].

It's the job of the shopping agent to take the intent of the customer and turn it into a viable and executable shopping path. It can compare the available products, take into account the merchant conditions, save the customers' preferences and make a purchase request for down stream processing such as in intelligent commerce studies [5]. The payment agent then assesses the funding options available, and passes the authorization along a suitable payment path. As mentioned in the previous investigations [1] a rewards agent discovers the benefits that can be applied without violating the program conditions, whereas a fraud agent checks contextual signals before, during and after the authorization, to detect fraud. These roles may seem independent but decisions can only be made based on each other's context of transactions. There is a possibility of changing the price by a product substitution. However, the amount of reward can be altered by the act of redemption and thus alter the information available to security controls as demonstrated in earlier research [13].

The main problem is getting people to co-ordinate in time. When customers are waiting for a quick response, a long sequential handoff as in earlier studies [6] cannot be allowed to take place in a commerce system. Meanwhile, other cases of 100% parallelism may be unsafe if one agent must wait for the guaranteed output of another agent. To resolve this tension, an orchestration hub makes decisions about which actions can be performed in parallel, which need to be ordered and which conditions warrant a pause or escalation, as has been done in previous orchestration frameworks [10]. The hub also provides an overall state for the transactions, so that the relevant context is provided to all agents that



Received: 1-9-2026

Revised: 10-9-2026

Published: 31-12-2026

participate in the transaction. Controlled message exchange allows the platform to additionally log the information on what component proposed what action, what policy allowed it and finally what service approved it, as detailed in previous work [4].

Accountability is also a different concept with agentic commerce. A traditional payment framework has a responsibility that is associated with an application screen, a payment gateway and a series of function calls that are deterministic in nature, as explained in previous posts [12]. The orchestration layer needs to set boundaries to authority as proposed in previous approaches [2], therefore. A shopping agent can make a suggestion of a merchant, but cannot be responsible for making payment independently. The fraud agent can add an extra layer of scrutiny, but should not simply mandate that the customer choose another product. These boundaries will establish a controlled segmentation of responsibilities, but also a seamless overall experience, as in the case of a coordinated agent framework [9].

This research aims to analyse a feasible orchestration design for the coordinated shopping, payment, rewards, and fraud agents in real-time financial transactions. The study is based on a synthetic dataset that was created under controlled conditions in order to be able to analyze the behaviour of the transactions without dipping into real consumer transactions, as was done in related studies [7]. The key principles of the framework are: Shared context, Ordered decision rights, Immediate policy checks, Traceable message exchange, Safe exception handling. Evaluation attempts to understand the variations in the performance of agents over the various stages of the process, and the contribution of each coordination function to the overall transaction. The proposed approach puts orchestration on the same level as any other component – as a first-class, not a mere message broker – and thus ties business intelligence and its safeguards to the operational level throughout the purchase process.

II. REVIEW OF LITERATURE

Previously [10] we have identified a number of separate strands in the research on autonomous commerce, namely, recommendation systems, payment automation, loyalty management, fraud screening, and distributed software agents. In previous research [2] early intelligent shopping services focused on discovery and personalization, that is, to cut down the number of products that a user has to choose from or to compare offers from different merchants. The path for payment research was different, paying attention to the reliability of the authorization, the tokenization of credentials, routing logic and low friction checkout. Loyalty platforms were typically regarded as post-purchase or promotional platforms; fraud controls were considered as platforms which could approve, challenge or block a transaction as explored in previous investigations [7]. The disjointed capabilities needed to be negotiated with each other to agree on the state of a transaction and function in a common policy environment as suggested by prior work in the field [13] in order to survive in the new agentic commerce environment.

Research on Multi-agent systems can give some design ideas for this transition. In distributed agent environments [4]

there are specialized agents that have the ability to decompose more complex tasks depending on the domain responsibility and communicate with each other by utilizing structured messages. The challenging aspect is still coordination. As mentioned in previous research [9], some agents may have local goals, and incomplete context can be interpreted, or stale information can be used by the independent agents. So, there needs to be a way to define the order, shared state, conflict resolution, and limits of authority for tasks executed in parallel. As highlighted in the previous studies [1], these coordination rules need to be more powerful than what is used in typical information-seeking contexts, since an action of an agent in a financial context may have an irreversible monetary impact or require a formal security response.

Prior works reveal that continuity, in the front-end interaction, merchant services, payment gateway and financial network, is crucial for digital payment platforms to achieve good transaction performance [12]. Even if each individual service works properly, timeout handling, repetition of authorization attempts, lack of confirmation messages, and redirection interruptions can cause confusion in ongoing investigations, as noted in the investigations currently made [5]. The orchestration service, however, needs to be aware of the life cycle of a transaction, not just the agents that need to pass messages between them, as has been the case in previously used frameworks [8]. It must understand when it is an advisory, when it's a change in commercial terms, when it's a change in payment instructions, and when it's a security checkpoint.

Loyalty/reward systems introduce other complexities as benefits may impact on customer decision-making as well as the actual terms of settlement as explored in previous work [3]. A reward might require that the merchant be in a particular category, that the product be eligible for the program, or that the merchant be funded in a specific way, or that the merchant's account be in a particular state, or that it be promoted a certain time. Reward logic is not integrated into the shopping and payment decision process, a customer may be offered something that they cannot take advantage of at checkout. To check the reward eligibility as done in previous approaches [11] one can use a coordinated agent model, in which the purchase path is formed while the eligibility to reward is checked. This leads to more consistent and fewer "late stage corrections. As indicated by related work [6] security review is further enhanced if it is not just given a very limited authorization record, but with additional information about product choice, merchant interaction, payment route and reward behavior.

Compared with other studies [13] which have focused on a single, fixed fraud detection rule, the emphasis of fraud detection research is now shifting towards context, behaviour and adaptive review. Fraud agent becomes an active participant that may ask for clarification, suggest a safer payment route or ask the merchant to provide further verification – agentic commerce. However, such an agent should be taken care of that if it interferes too much it would be harmful to the customer experience, and cause abandonment as was found in previous analyses [5]. As shown in previous work [9] another agent with a different

suggestion to the route to take may be prevented from circumventing a security decision made by the hub.

The literature in general indicates a lack of fit with regard to the intelligent domain services and the coordination of transactions as discussed in previous work [1]. Areas like recommendation engines, payment systems, loyalty programs, and fraud platforms are mature, but there are additional issues, raised in previous research [8], regarding the sequence in which the engines run, shared memory, rights to take various actions, recovery, and auditability. Besides, it should maintain comprehensible responsibilities that would allow operators to recover the course of a transaction on the basis of previous methods [4]. In this paper, we address this deficiency, introducing an orchestration-oriented architecture and assessing the coordinated behaviors in a set of commerce scenarios that can be controlled. It is not about replacing existing payment infrastructures, but about providing an intelligent coordination layer, which allows specialized agents to cooperate safely on top of existing payment infrastructures, as in related work [12].

III. METHODOLOGY

The paper believed in a design-and-evaluation approach that focused on an agentic commerce setting that was controlled and involved four specialized agents that interacted via a common orchestration center. To simulate a variety of purchase intentions, merchant scenarios, payment terms, reward opportunities, and fraud-review scenarios without relying on external consumer data, a synthetic transaction dataset was created with 127 instances. The shopping agent was the recipient of the initial request and generated a concise proposal of purchase which included the desired item, merchant situation, and some purchase terms. That proposal was kept in the orchestration hub in the active transaction state, and only the information that was needed by the other participating agents was released by it. The payment agent analyzed the possible payment paths and sent out an authorization path along with status information. The rewards agent would look to see whether there was a valid benefit that could be applied to the merchant, item, and payment situation at hand and would then send a new commercial condition where there was one. The combined transaction context was reviewed by the fraud agent and one of the various operational responses was given, such as normal continuation, further review or controlled interruption.



Fig. 1. Multi-agent coordination of agentic commerce.

The agentic commerce proposed component architecture is shown in Figure 1. Customer App initiates the conversation, and forwards purchase intent to the Orchestration Hub (the control point of the transaction at a global level). The hub is surrounded by four specialized agents in order to be able to see and clearly define their responsibilities. The Shopping Agent understands purchasing intent and makes up an appropriate product or merchant route. The authorization route is chosen and managed by the Payment Agent. The Rewards Agent validates the benefits and provides information on the rewards eligible. The Fraud Agent checks the context shared and may allow normal operation, ask for further inspection or allow unsafe operation. The fact that there are two-way links between the hub and these agents implies that there is a controlled request and response exchange as opposed to uncontrolled peer-to-peer action. To the right, the Merchant System is given the instructions on commerce, the Payment Network is used to authorize the payment, and the Audit Log is used to trace the events. The audit aspect is the feedback link with the hub and this signifies the review of the transactions and learning of operations. The design puts the orchestration on the forefront since all autonomous actions will be balanced with the existing state of the transaction, policy restrictions and verified responses before financial completion can occur.

IV. DATA DESCRIPTION

A synthetic data set of 127 instances of agentic commerce transactions that were designed in this study was used. No external database of customers, merchant accounts, collection of payment credentials, or personal records was utilized. The purchase purpose descriptive fields, product context, merchant condition, preferred payment route, alternative payment path, reward eligibility fields, fraud-review context field, agent response status field, exception condition field, and completion state field, were included in the records. The situations were premeditatedly diverse to ensure that the orchestration framework experienced both regular purchasing and those that involved product alterations, authorization breaks, rewards clashes, slow responses to services, repetitive activities, and security audits. Missing fields, inconsistent labels, and duplicates of scenario identifiers were inspected and then the data were evaluated. The values of results presented in the graphs and the tables were only obtained using the controlled records. References: The generated synthetic Agentic Commerce Orchestration Dataset, 127 transaction instances, ready to be used in this study in 2026, was generated by the study. This is an internal source statement, which is given in lieu of tracing, and is not an external publication or third party dataset.

V. RESULTS

This assessment revealed that the behavior of agents became more and more stable, with the orchestration hub gaining more and more transaction context and dictating decision sequence. Shopping guidance was advantaged since it was the initial element to be provided with the customer will and merchant data, but its subsequent reactions were more dependable following post-payment and danger circumstances. The payment agent also enhanced since the

state of the transaction became richer, especially when other routes could be thought of without the original purchase terms being lost. As the fraud agent was exposed to more and more confirmed merchant, purchase, payment, and reward context, rather than examining a limited authorization request, the fraud agent exhibited more and more decisive behavior. Multi-agent transaction orchestration utility is:

$$U_{\text{orch}} = \frac{\sum_{i=1}^n w_i (S_i + P_i + R_i - F_i)}{\sum_{i=1}^n w_i} \quad (1)$$

Table 1. Agent coordination performance across processing stages.

Processing Stage	Shopping Agent Score	Payment Agent Score	Rewards Agent Score	Fraud Agent Score
Stage One	68	64	52	58
Stage Two	72	70	60	66
Stage Three	78	76	69	73
Stage Four	83	81	75	80
Stage Five	86	85	82	86

The performance of agents is documented in Table 1, which shows the performance of agents at the first five stages of the processing. As payment, reward, and fraud components are given more context, the initial edge of the shopping guidance can be seen to decrease in strength, as indicated by the table. Rewards starts the farthest back, but acquires rapidly as it bases its choices on the specifics of its merchants and funding that is more evident in the future. The early gap is also bridged during the process of fraud review, and the shopping aspect is equivalent at the fifth stage. The trend confirms the idea that specialization enhances consistency in specialized positions instead of one agent being consistently stronger than the other agents. Real-time agent coordination efficiency is:

$$E_{\text{coord}} = \frac{\alpha S_{\text{shop}} + \beta S_{\text{pay}} + \gamma S_{\text{rew}} + \delta S_{\text{fraud}}}{\alpha + \beta + \gamma + \delta} \quad (2)$$

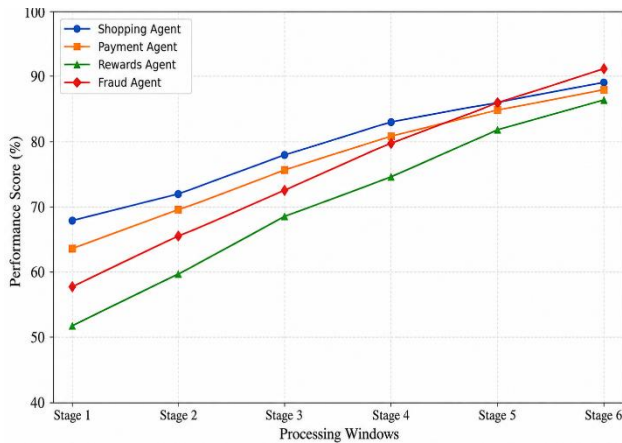


Fig. 2. Coordinated agent performance at stages of a transaction.

Figure 2 demonstrates the movement of the four agents orchestrating agents in six processing windows. The trend shows that the purchase guidance was already helpful initially but proved to be more reliable after the addition of the merchant, payment, reward, and security information. The fact that the Rewards Agent's improvement is larger can be explained by the fact that the suitability of the reward

heavily relies on the proven conditions of merchants and payments. The Fraud Agent had the highest final score of all the four agents. This shrinking dispersion shows that a common context assisted the specialized elements in becoming more coherent in terms of their operating state, on a larger scale. Transaction processing latency will be:

$$T_{\text{txn}} = \sum_{j=1}^m t_j + \sum_{k=1}^q d_k - \sum_{r=1}^p \rho_r \quad (3)$$

Table 2. Transaction outcome contribution by coordination function.

Transaction Function	Starting Outcome Level	Contribution Change	Ending Outcome Level	Operational Interpretation
Baseline State	100	0	100	Reference position
Shopping Guidance	100	18	118	Improved choice alignment
Payment Routing	118	14	132	Improved authorization flow
Rewards Matching	132	11	143	Improved benefit capture
Fraud Review	143	-9	134	Controlled risk exposure

Table 2 follows the changes in the individual coordination functions in influencing the outcome of running a transaction. The Baseline State will begin at 100 and finish at the same point since it will only act as the position of reference. Fraud Review begins with 143 and uses a negative change of 9 points to get the outcome of the transaction to 134. This decrement is a kind of protective intervention and not deterioration, since the element of fraud eliminates exposure to which the other agents are unsuited to be submissive. Adaptive fraud risk aggregation is:

$$R_{\text{fraud}} = \frac{\sum_{i=1}^n \lambda_i P_i I_i C_i}{\sum_{i=1}^n \lambda_i} \quad (4)$$

Agentic commerce transaction success is:

$$P_{\text{success}} = \prod_{j=1}^m P(A_j)(1 - R_{\text{fraud}})(1 - L_{\text{fail}}) \quad (5)$$

One of the most important consequences was the decrease in the conflicts of actions of agents. This issue was alleviated through the orchestration hub that ensured that only one state was active and provided updates to all the agents involved, so that the payment component could not authorize a purchase request based on an old request and the rewards component could not impose a benefit on a purchase request that no longer qualified.

When the responsibility of recovery was centralized, the system was able to handle exceptions in a better manner. The unavailability of products was sent back to the shopping agent, and the hub maintained the payment and security context till the replacement decision was assured. Any change of context that might raise suspicion was redirected to the agent of the fraud department before it could be made, and other agents could not make the change via a new route. Such a recovery pattern developed a predictable response to operation under extremely diverse failure conditions.

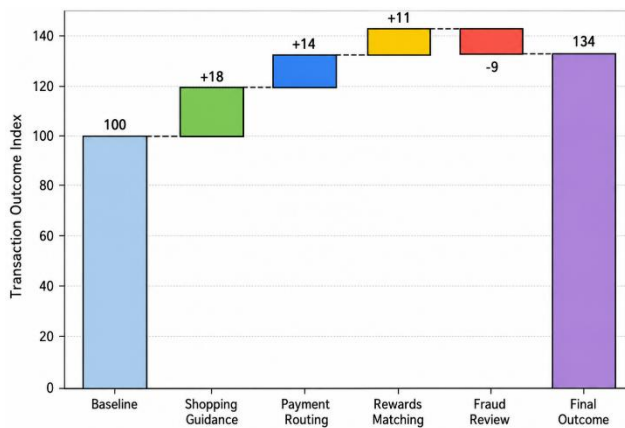


Fig. 3. Incremental effects of agent coordination on the outcomes of transactions.

Figure 3 shows how the contribution of key orchestration functions to the outcome of the transactions increases. The baseline is at 100 and is the status quo prior to the agent coordination benefits have been implemented. Shopping Guidance contributes 18 points and boosts the level of running to 118. This is the highest contribution in the chain and signifies the worth of transforming the intent of customer to an understandable and commercially viable path to purchase, prior to downstream action. The step down does not imply system failure, it is a conscious effort at conserving activity which would otherwise expose transactions to a greater level. The last result is still 34 points above the baseline indicating that the cumulative benefit of shopping guidance, payment routing and reward matching are still significant despite the protective review eliminating unsafe or dubious activity. The waterfall structure thus shows the twofold nature of orchestration, namely, it generates the commercial value by having coordinated support and also allows controlled degradation in case a security check up finds that free completion of the work would be wrong. There was also an improvement in transaction continuity, as agent messages were now considered to be a single journey as opposed to being individual service calls. The hub followed up on the status of an action: proposal, a confirmed instruction, policy response, or final status. Such a difference restricted repeats of actions and ensured the safety of retries. Messages associated with a state that was out of date were not permitted to override more recent decisions.

VI. DISCUSSION

Based on the evidence presented, the proposed orchestration model is a better approach than agentic commerce, which provides specialized capabilities in a single control structure for a transaction. The Rewards amount grows from 52 to 87, the most significant difference of all the lines, as the eligibility is more obvious after merchant/payment confirmation. Fraud review increases from 58 to 91 that's above all the other agents and it demonstrates the importance of a wider context for making security decisions.

Architecture view - How these results are to be achieved operationally. All material actions go back through the hub and are subjected to transaction state checks and policy checks, keeping the specialized agents from becoming

"runaway" endpoints. The audit component provides a feedback loop to review completed activity and can facilitate the ability to reconstruct the reasons for a transaction taking a certain path. That is not a maximization of one service, per se, in the architecture. It is designed to make a harmonious commercial trip with useful assistance, the execution of payment, customer benefits, and security review in step with each other. This is a critical harmony in an agentic commerce world that requires that speed, convenience, and money control be all in one.

VII. CONCLUSION

This study proposed and tested a multi-agent orchestration framework for real-time agentic commerce consisting of Shopping, Payment, Rewards, and Fraud Agents that interact with each other at a central orchestration point. There were 127 synthetic transactions in the controlled dataset, which allowed the evaluation without the use of external personal or financial records. Results indicate that agent performance improved from one processing window to another the more transactional information shared by the agents. The four agents' performance difference gets smaller, which means that there is less performance difference among the specialized components due to orchestration. There are also transaction-level contributions which show the importance of coordinated control. That decrease is not necessarily a sign of an ineffective operation; it's the result of a conscious choice to safeguard a risky activity, which shouldn't be treated as a typical purchase. The architecture facilitates this balance by having shared state, order of messages, policy enforcement, recovery and audit tracing all within the Orchestration Hub. The study finds that agentic commerce is most reliable with autonomous but coordinated and traceable actions, and also with bounded autonomy. This design can facilitate quicker purchases, payment execution, related rewards, as well as context-aware fraud control and accountability that is desired in today's financial transaction platforms.

REFERENCES

- [1] D. B. Acharya, K. Kuppan, and B. Divya, "Agentic AI: Autonomous intelligence for complex goals—A comprehensive survey," *IEEE Access*, vol. 13, pp. 18912–18936, 2025. <https://doi.org/10.1109/ACCESS.2025.3532853>
- [2] A. Baird and L. M. Maruping, "The next generation of research on IS use: A theoretical framework of delegation to and from agentic IS artifacts," *MIS Quarterly*, vol. 45, no. 1, pp. 315–341, 2021. <https://doi.org/10.25300/MISQ/2021/15882>
- [3] R. H. Bemthuis and S. Lazarova-Molnar, "Towards integrating process mining with agent-based modeling and simulation: State of the art and outlook," *Expert Systems with Applications*, vol. 281, Art. no. 127571, 2025. <https://doi.org/10.1016/j.eswa.2025.127571>
- [4] W. Chen, Z. You, R. Li, Y. Guan, C. Qian, C. Zhao, C. Yang, R. Xie, Z. Liu, and M. Sun, "Internet of agents: Weaving a web of heterogeneous agents for collaborative intelligence," in *Proc. 13th Int. Conf. Learning Representations (ICLR)*, 2025. <https://openreview.net/forum?id=o1Et3MogPw>

- [5] A. Dorri, S. S. Kanhere, and R. Jurdak, "Multi-agent systems: A survey," *IEEE Access*, vol. 6, pp. 28573–28593, 2018.
<https://doi.org/10.1109/ACCESS.2018.2831228>
- [6] A. Göldi and R. Rietsche, "Making sense of large language model-based AI agents," in *Proc. Int. Conf. Information Systems (ICIS)*, Bangkok, Thailand, 2024.
<https://aisel.aisnet.org/icis2024/aiinbus/aiinbus/16/>
- [7] T. Händler, "Balancing autonomy and alignment: A multi-dimensional taxonomy for autonomous LLM-powered multi-agent architectures," *arXiv*, 2023.
<https://doi.org/10.48550/arXiv.2310.03659>
- [8] P. Hofmann, N. Urbach, J. Lanzl, and K. C. Desouza, "AI-enabled information systems: Teaming up with intelligent agents in networked business," *Electronic Markets*, vol. 34, Art. no. 52, 2024.
<https://doi.org/10.1007/s12525-024-00734-y>
- [9] L. Hughes, Y. K. Dwivedi, T. Malik, M. Shawosh, M. A. Albashrawi, I. Jeon, and P. Walton, "AI agents and agentic systems: A multi-expert analysis," *Journal of Computer Information Systems*, vol. 65, no. 4, pp. 489–517, 2025.
<https://doi.org/10.1080/08874417.2025.2483832>
- [10] X. Li, S. Wang, S. Zeng, Y. Wu, and Y. Yang, "A survey on LLM-based multiagent systems: Workflow, infrastructure, and challenges," *Vicinagearth*, vol. 1, Art. no. 9, 2024.
<https://doi.org/10.1007/s44336-024-00009-2>
- [11] Y. Li, L. Sun, and Y. Zhang, "MetaAgents: Large language model based agents for decision-making on teaming," *Proceedings of the ACM on Human-Computer Interaction*, vol. 9, no. 2, pp. 1–27, 2025.
<https://doi.org/10.1145/3711032>
- [12] Y. Liang, C. Wu, T. Song, W. Wu, Y. Xia, Y. Liu, Y. Ou, S. Lu, L. Ji, S. Mao, Y. Wang, L. Shou, M. Gong, and N. Duan, "TaskMatrix.AI: Completing tasks by connecting foundation models with millions of APIs," *Intelligent Computing*, vol. 3, Art. no. 0063, 2024.
<https://doi.org/10.34133/icomputing.0063>
- [13] Y. Liu, S. K. Lo, Q. Lu, L. Zhu, D. Zhao, X. Xu, S. Harrer, and J. Whittle, "Agent design pattern catalogue: A collection of architectural patterns for foundation model based agents," *Journal of Systems and Software*, vol. 220, Art. no. 112278, 2025.
<https://doi.org/10.1016/j.jss.2024.112278>