

Concept of an Agent-Based, Distributed Web Application Utilizing AI for forecasting Startup Profitability*

Natalia PAWŁOWSKA and Ewa RATAJCZAK-ROPEL

Gdynia Maritime University, Gdynia, Poland

Correspondence should be addressed to: Natalia PAWŁOWSKA, n.pawlowska@wi.umg.edu.pl

* Presented at the 46th IBIMA International Conference, 26-27 November 2025, Ronda, Spain

Abstract

This paper presents the concept of an agent-based, distributed web application utilizing artificial intelligence (AI) for forecasting startup profitability. The proposed system integrates key success factors (KSFs) to evaluate financial and operational performance across different stages of a startup's life cycle. The application employs autonomous agents that collect, process, and analyze multidimensional data to generate profitability forecasts and recommendations. The multi-agent architecture ensures scalability, adaptability, and resilience through asynchronous communication and self-organizing behaviours. Seven types of agents are defined performing specialized tasks to enable collaborative learning and dynamic decision-making. The system design allows further extensibility by adding new agents. This concept provides a foundation for the development of intelligent decision-support tools that assist entrepreneurs and investors in making data-driven strategic choices throughout a startup's lifecycle.

Keywords: multi agent system (MAS), AI agent, startup profitability, data analysis.

Introduction

With technological advancement and the increasing volume and speed of information transfer, new products and services are emerging, along with a form of business activity referred to as start-ups. They are described as temporary organizations whose goal is to identify a scalable, repeatable, and profitable business model (Blank and Dorf, 2020). Start-ups are enterprises less than ten years old that possess innovative technologies and business models, and strive for an increase in the number of employees or in sales growth (Kollmann *et al.*, 2015).

The assessment of a startup's profitability is a crucial stage both during its establishment and throughout its entire life cycle. Such information is essential for both entrepreneurs and investors. For startup founders, understanding profitability is vital for effective resource management, planning, and development. Investors, in turn, require this information to make informed decisions, conduct negotiations, anticipate potential challenges, and maximize return on investment.

Forecasting the profitability of enterprises, particularly start-ups, is a complex and challenging task, yet it is of critical financial importance. Numerous methodological approaches to this problem have been described in the literature. When start-ups are treated as any other business ventures, it can be observed that the primary factor determining a firm's survival or contributing to its bankruptcy is its financial data (e.g., income, assets, share capital) (Becchetti and Sierra, 2003). The credibility of a given business venture may, for instance, be assessed based on the level of its share capital. Furthermore, it is possible to identify factors that determine the activities

of these businesses, which are external in nature and result from the activities of start-ups in their surrounding ecosystems. These factors may include efficiency, financing, market reach, talent and experience, AI-Native transition, and knowledge (The Global Startup Ecosystem Report 2025). Another approach involves the analysis of key success factors (KSFs). The KSF concept is based on the assumption that, among numerous factors, only a few determine an enterprise’s competitive position, as well as its success or failure. A set of key success factors comprises a list of characteristics that are necessary, though not sufficient, to achieve success.

Multi-agent systems (MAS) are widely used in different areas. They use multiple autonomous agents that cooperate, communicate, or compete to solve difficult problems. Enhanced with modern AI—especially machine learning and reasoning models — these systems can support business or other activities (Li *et al.*, 2024). In data analysis, AI-driven agents can divide large datasets, explore different analytical strategies in parallel, and fuse their results to improve accuracy and robustness. This makes MAS particularly valuable for distributed analytics, real-time decision-making, and environments where information is incomplete or arrives asynchronously.

In this paper a concept of distributed multi-agent web application that uses KSF and AI to predict a startup’s future profitability, cash flow, and growth. It autonomously gathers financial and operational data, processes it through intelligent agents, and produces probabilistic forecasts, and recommendations. The multi-agent architecture ensures scalability, resilience, and real-time adaptability — allowing founders and investors to make data-driven strategic decisions.

Identifying key success factors for startups

The issue of startups is directly related to their life cycle, which consists of five stages. The first of these is pre-seed stage, during which the idea for a business venture is conceived. The next is the seed phase, during which the business idea begins to take shape, typically in the form of a business plan. The following phase, known as early growth (or early stage), refers to the period after the business idea has been validated. It involves new, significant financial investments and attempts to establish a stable market position. During this phase, both revenue and the pace of development become relatively stable. In the expansion phase, the startup transforms into a mature company. Its market position, income, and growth rate are relatively consistent. The final phase of the life cycle, referred to as the late stage or late growth phase, concerns the period during which the owner seeks opportunities for further business expansion and once again evaluates its profitability.

In each of this stage different key success factors (KSF) describe the startup profitability. Based on the earliest research (Mańkowska and Kamińska, 2021), for the needs of the proposed application design, 8 key success factors for start-ups are identified:

- Founding Team,
- Market Timing,
- Product-Market Fit (PMF),
- Business Model and Revenue,
- Technology and Competitive Advantage,
- Finance and Operational Efficiency,
- Ecosystem and Support Network,
- Timing and Macro Trends.

For each of the KSFs the indicators and measures are chosen to calculate the predicting the profitability in each life cycle stage. Examples of these indicators for 3 first KSFs are presented in tables 1-3.

Table 1. Founding team indicators

Stage	Measured	Indicators/measures
pre-seed/seed	Experience of the founders	Years of experience, earliest start-ups, exit history, network score
seed/growth	Team dynamics	Retention of key people, employee growth rate
growth/expansion	Management efficiency	venue per employee, employee churn

Table 2. Market timing

Stage	Measured	Indicators/measures
seed	Market size	Total Addressable Market (TAM), SAM, SOM
growth	Market growth	Compound Annual Growth Rate (CAGR), number of customers in the segment
expansion	Market share	market share, year-over-year growth vs competition

Table 3. Product-Market Fit

Stage	Measured	Indicators/measures
seed	Product Adoption Rate (PAR)	Number of active users (DAU, MAU), retention rate
growth	Customer satisfaction	Net promoter score (NPS)
expansion	Product scalability	the cost of acquiring and servicing a user, uptime, capacity

The proposed indicators and measures are used by the Analytics Agent described in the next sections.

Application architecture concept

In this paper a concept of distributed multi-agent web application that uses KSF and AI to predict a startup's future profitability, cash flow, and growth is proposed. The application is scalable, distributed, agent-base system based on the concept of mikroservices (Newman, 2021), multi-agent system (Dorri, Kanhere and Jurdak, 2018), asynchronous team idea (Talukdar *et al.*, 1996) and AI agents (Li *et al.*, 2024). In this system specialized autonomous agents cooperate to:

- Collect, clean and validate data from multiple sources,
- Choose AI agents and/or ML models,
- Engineer features and learn models,
- Forecast start-up profitability based on related KSFs,
- Explain results and adapt their behaviour over time.

Agents communicate via a message bus (event-driven) and coordinate through a multi-agent orchestration layer. Each agent is autonomous but follows shared protocols for data exchange, negotiation, and coordination. Agents cooperate together to complete the task and solve the problem. (Ratajczak-Ropel, 2018; Jędrzejowicz and Ratajczak-Ropel, 2022).

The code of the application is divided to 4 Layers: Client UI Layer, API Gateway, Multi-Agent Layer, Distributed Data Layer. There are 7 kinds of autonomous agents. Two of them work in API Gateway: Orchestrator Agent and Data Collecting Agent. The remaining ones: Data Processing Agent, Analytics Agent, AI Model Training Agent, AI Forecasting Agent and Monitoring Agent work in Multi-Agent Layer. All agents can be duplicated to scale the application. The general schema of the architecture is presented in Figure 1.

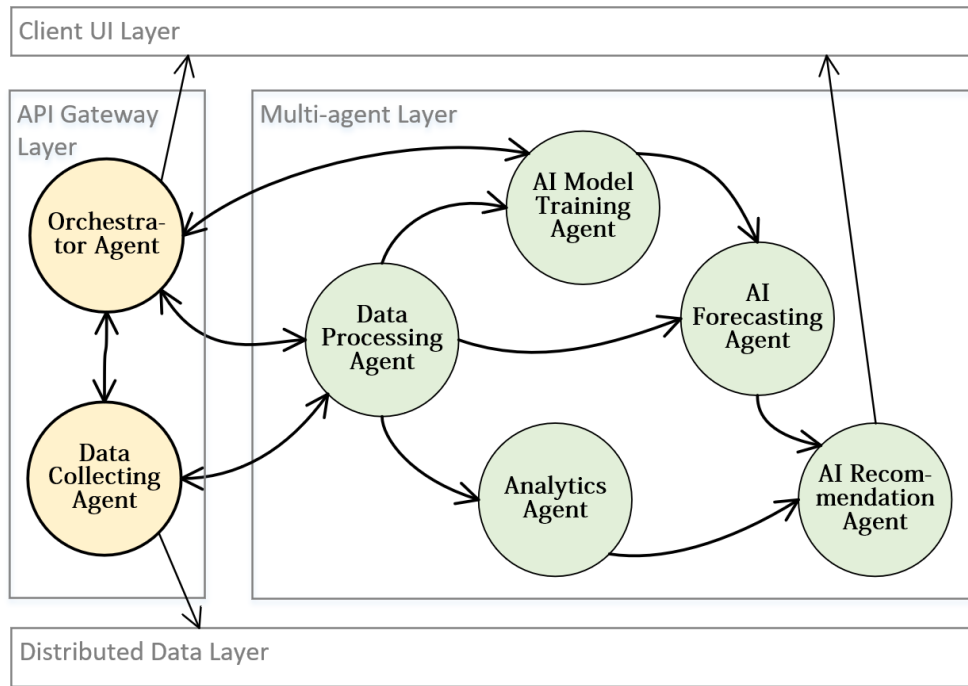


Fig. 1. General schema of the application architecture.

Each agent in this multi-agent system (MAS) operates independently, performs a specific role within the architecture, and communicates with other agents. This system relies on decentralized control, allowing agents to coordinate their actions without the need for a central authority. This distributed structure increases robustness, flexibility, and scalability, enabling the system to adapt dynamically to changing environmental conditions. In addition, cooperation and negotiation among agents contribute to more efficient problem-solving and improved overall system performance.

Agents and their roles in the system

The **Orchestrator Agent** is a coordinator of the application running. It is a central conductor that manages lifecycle, workflow orchestration, and delegation of tasks to other agents. It authenticates and authorizes incoming user/API requests. It can decompose high-level tasks into subtasks for downstream agents and maintain global state and negotiation between agents. It is responsible for routing messages via ACL (Access Control List). The Orchestrator Agent also manages error recovery and fallback strategies. It receives requests from API Gateway, sends work to other agents and finally aggregates and returns results.

The **Data Collecting Agent** autonomously searches and connects to external data sources and maintains data freshness. Such an agent can be universal or specialized in collecting only one kind of data, for example Finance Data Collecting Agent or Market Data Collecting Agent. Data Collecting Agent schedules or triggers data ingestion, validates schema and completeness of the data, publishes normalized events to message bus, collaborates with Data Processing Agents. It produces validated data streams.

The **Data Processing Agents** receive data streams from Data Collecting Agents and are mainly responsible for feature engineering. Data Processing Agent transforms raw data streams into model-ready features. It cleans and completes the data and ensures consistency. Data Processing Agent sends the processed data to the Data Layer.

The data from Data Layer are received by **Analytics Agent**. It computes indicators and metrics described earlier in section Identifying key success factors for start-ups. The Analytics Agent is responsible for sending the produced data to Data Layer. It is possible to add additional analytical agents using other measures and indicators or using external LLM models.

The **AI Model Training Agent** owns the lifecycle of AI/ML models — training, retraining, versioning, and validation. It subscribes to feature updates from Data Layer, trains distributed ML models, evaluates backtests (MAE, CRPS, calibration), pushes trained model artifacts to the Model Registry in Data Layer, notifies Forecasting Agent when new models are deployed.

The **AI Forecasting Agent** generates profitability forecasts and financial projections using ensemble models. It pulls models from Model Registry, gets features, receives scenario requests from Orchestrator Agent and publishes predictions, confidence intervals, and explanations. It can use statistical and machine learning models of forecasting.

The **AI Recommendation Agent** gathers results, interprets AI predictions and generates human-readable insights. It generate natural language explanations (using an LLM), send visual and textual summaries to Client UI Layer. It can collaborate with Forecasting Agent to adjust model interpretability settings dynamically. It uses causal and simulation outputs to recommend strategic actions. This agent identify potential interventions, optimize using multi-objective optimization or reinforcement learning for prescriptive guidance.

Additionally, the **Monitoring Agent** is used to ensure reliability, ethical behaviour, and adaptation of the multi-agent system. It detects data or model drift and signals retraining, monitors for unauthorized access or anomalous data behaviour, oversees retraining triggers, version control, and rollbacks.

All agents cooperate according to the defined strategy to ensure the best results. The details of the cooperation strategy are not described in this paper.

Conclusions

In this paper, the concept of agent-based, distributed web application utilizing AI for forecasting start-up profitability has been proposed. The agents operate independently and can adapt to the environment through self-tuning, retraining or task delegation. For the purposes of the application, key success factors are identified and adopt in the system. The AI capabilities are used to training models, forecasting and generating recommendations. This concept is a foundation for the development of intelligent decision-support system that supports in making data-driven strategic decisions throughout a start-up's lifecycle.

The natural continuation of this work will involve the expansion and refinement of the design, selecting appropriate technologies, implementing the system and conducting tests.

The proposed architecture can be extended by duplicating agents, but also by introducing additional ones. For example, a Learning Agent could be added to monitor the performance of other agents, optimize hyperparameters, or improve learning strategies. Moreover, a Negotiation Agent could be implemented to allocate computational resources and balance workloads.

Bibliography

- Becchetti, L. and Sierra, J. (2003) 'Bankruptcy risk and productive efficiency in manufacturing firms'. *Journal of Banking & Finance*, 27(11).
- Blank, S. and Dorf, B. (2020) 'The Startup Owner's Manual. The Step-by-Step Guide for Building a Great Company', John Wiley & Sons Inc.
- Dorri, A., Kanhere, S. S. and Jurdak, R. (2018) 'Multi-Agent Systems: A Survey', in *IEEE Access*, vol. 6, 28573-28593.
- Jędrzejowicz P. and Ratajczak-Ropel, E (2022) 'A-Team Solving Multi-Skill Resource-Constrained Project Scheduling Problem', *Procedia Computer Science*, (207), Elsevier, 3294-3303, <https://doi.org/10.1016/j.procs.2022.09.388>.
- Kollmann T., Stöckmann C., Linstaedt J. and Kensbock J. (2015) *European Startup Monitor 2015*, <http://europeanstartupmonitor.com/esm/esm-2015/>.
- Li, X., Wang, S., Zeng, S. et al. (2024) A survey on LLM-based multi-agent systems: workflow, infrastructure, and challenges. *Vicinagearth* 1, 9, <https://doi.org/10.1007/s44336-024-00009-2>
- Mańkowska, N. and Kamińska, K. (2021) 'A description of the development of HR Tech startups in Poland', *Scientific Journal of Gdynia Maritime University*, (118), 38-50.
- Newman, S. (2021) *Building Microservices: Designing Fine-Grained System*. O'Reilly Media.
- Ratajczak-Ropel, E. (2018) Agent-based approach to the single and multi-mode resource-constrained project scheduling, *Studies in Systems, Decision and Control*, (108), 3–100, <https://doi.org/10.1007/978-3-319-62893-6>.
- Talukdar S., Baerentzen L., Gove A. and De Souza P. (1996) *Asynchronous teams: cooperation schemes for autonomous, computer-based agents*. Technical report EDRC 18–59-96, Carnegie Mellon University, Pittsburgh.

- The Global Startup Ecosystem Report 2025 (GSER 2025), (2025), *Startup Genome*, <https://startupgenome.com/report/gser2025/introduction>.