INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT (IJSREM)
SPECIAL EDITION - VOLUME 09 NCFT -2025 SJIF RATING: 8.586 ISSN: 2582-3930

Replacing AI Agents for Backend

Basaveni Siri Mallika Rao[1]
Computer science and engineering
Sri Indu Institute of engineering and technology
Hyderabad, India
sirimallikarao.basaveni@gmail.com

Sai Santosh Goud Bandari[2]
Software Engineer
Tata Consultancy Services
Nc,Raleigh
bandari.santhosh007@gmail.com

Abstract—Creating modern applications using a heavy backend is a big task for developers. Front end can be created easily using scripting languages like HTML, CSS, java script, react frameworks etc. and many low code tools like WordPress, Figma, Bolt etc., but creating a fully working backend is a difficult part for developers, students or startups. Instead of writing thousands of lines of code we can build applications using AI agents. Sounds unreal right? But yes, it's possible we can replace maximum of our backend logic using AI agents.

Traditional backend development involves significant manual effort in writing, testing and maintaining code to support functionalities of backend such as authentication, database management, business logic and notifications.

With emergence of AI and workflow orchestration platforms, we can see a high potential on how we can transform backend systems using these intelligent agents, with advancements in artificial intelligence and no-code orchestration platforms such as n8n, there is a drastic change where backend systems can be created, managed and evolved by AI agents.

This paper explores how AI agents can transform backend development eliminating boilerplate code and introducing adaptive, scalable and intelligent architectures to design an application.

Keywords—*AI Agents, Backend Development, Workflow Automation, n8n, No-Code Platforms, Intelligent Systems, Application Architecture, Natural Language Processing, Low-Code Development, Large Language Models (LLMs), Orchestration Tools, Business Logic Automation, API Integration, Smart Workflows*

I. INTRODUCTION

Backend systems are like invisible workhorses of modern applications. They manage crucial functions in an application like user authentication, data processing [1], data storage, notification services, business logic execution and external API communication. Traditionally building these systems has been a complex task demanding significant coding languages like python, java, C++, .Net and many more core languages. It also involves setting up infrastructure, designing, databases, integrating various software components [2] or modules required to build an application and ensuring security.

II. EXISTING SYSTEM

Even though backend development has come a long way over the past few decades it still throws us some familiar challenges. High Development time where developers must write boilerplate code [3] for basic functionalities like CRUD operations, user management, email triggers etc. These applications also require complex maintenance like debugging backend logic [4] which often requires understanding deep system dependencies and logs. Scalability challenges while designing the systems which scale with user growth requires complex architecture and DevOps support [5]. Inflexibility issues like updating business logic typically demands code changes, testing and redeployment creating a lot of time delay.

Non-technical stakeholders or early-stage startups struggle to prototype apps due to backend complexity. These limitations result slow iteration cycles and increased costs especially in fast-paced environments like agile or early-stage environments, startups and hackathons.

Recent advancements in natural language processing and machine learning allows large language models (LLMs) [6] to comprehend and generate backend logic on demand. When these models are combined with no-code automation tools like n8n, these models can take required user inputs and convert them into executable workflows.

AI agents in this context are intelligent services collaborated by LLMs which can interpret user intentions from natural language, generate code or logic for building applications, interact with databases, APIs and authentication systems. They learn and adapt overtime based on usage patterns. We can modify the created workflows based on performance or feedback.

III. WORKFLOW ORCHESTRATION WITH N8N:

n8n [7] is an open-source node-based workflow automation tool which supports visual orchestration of backend logic. In an AI driven system, it can trigger nodes activate workflows based on HTTP requests or schedules. Logical nodes contain AI generated rules and regulations. Database nodes handle CRUD operations effectively. API and communication nodes connect with external platforms. Conditional branches and loops allow creating decision trees to make necessary decisions based on inputs and processes.

IV. REPLACING BACKEND COMPONENTS WITH AI:

The integration of AI agents into a backend system represents a transformative shift in how applications are built and maintained. By using AI, organizations can streamline processes, reduce manual coding and enhance overall efficiency in backend operations
A thorough examination of how AI agents [8] and coordinated workflows can be used to replace each significant backend function can be found below. The following backend elements can be fully or partially replaced by these AI agents:

A. Authentication and Authorization:

AI agents can transform the way user authentication and authorization is handled. These agents can handle user sessions, generate secure tokens, verify credentials and manage role-based access. By utilizing prebuilt authentication nodes in orchestration platforms like n8n, developers can start complex authentication flow with simpler and easier prompts like “allow only admins to access this particular data” and it will automatically configure the necessary OAuth flows and required permissions. This approach makes sure that security measures are not just deployed easily but also guaranteed that they protect the application consistently.

B. Database Management:

AI agents can significantly improve database management by translating natural language queries based on user instructions into SQL or NoSQL statements. Instead of having to learn complicated database languages this feature helps people work with the information they need for their task. These AI almost act like a translator so that you can talk to a database without learning its specific language. For example, if a user inputs AI agent, “Can you get me the details of all users who signed up in past week and email them a discounted code for new users.” This AI agent figures out technical stuff to find those users, gets their email addresses and send out the codes all without anyone having to write complicated queries. This makes it easy to work with information and get things done immediately based on user needs.

C. Business Logic Execution:

Usually, business logic is hardcoded into applications making it rigid and challenging to adapt. These AI agents can generate and modify business rules in real-time based on user interaction and feedback. Instead of hardcoding conditional rules or workflows these agents can generate decision trees and logical flows based on user description. For example, if an agent is instructed to execute a workflow that states “if the order value is >Rs.1000 then apply a 10% discount and notify to the warehouse.” Instead of depending on developers to write and redeploy code for each and every change, the AI agents can adapt the logic spontaneously allowing businesses to respond as quickly as possible to changing conditions and their requirement.

D. Notification Systems:

AI agents can manage complex notification workflows across various channels which includes email, SMS, push notifications and messaging platforms like Slack. By using environment-aware templates and decision logics these agents can figure out the most appropriate communication methods based on user preferences or specific triggers. For instance, if a user opts for SMS notification for urgent updates, then the AI agent can automatically forward messages or transmit messages accordingly ensuring that users receive immediate and relevant information without manual involvement.

E. Session and State Management:

While creating an application state and session management play a crucial role in maintaining user interactions and data across multiple requests ensuring smooth and personalized experience especially in web applications where HTTP protocol is stateless by nature. State management is necessary for providing a seamless user experience especially in applications that require user sessions. AI agents can make use of session-based memory or external state stores such as Redis or vector databases to remember user interactions and preferences for stateful experiences. This ability allows applications to maintain context across various sessions enabling personalized experience for users. For example, an AI agent could remember a user’s last viewed product or preferences of the user making it easier to provide customized or personalized recommendations for future interactions.

F. API Integration:

Integrating third-party APIs is often a complex and time-consuming task for developers so we can use AI agents which can simplify this process by generating HTTP requests with the necessary headers and payloads based on user instructions. For example, a user might request to “process a payment using payment gateway”, and the AI agent would automatically build the necessary API call to handle authentication and manage responses. These agents will erase the need for developers to write extensive integration code allowing a more faster and efficient API integrations.

G. Monitoring and Feedback:

AI agents play a crucial role in monitoring an application’s performance and providing necessary feedback for continuous improvements. By parsing logs and analyzing performance metrics in real-time, these agents can pinpoint or identify slow workflows, detecting anomalies and recommend required optimizations based on historical behavior. If an AI agent notices that any particular API call in an application is constantly slow then these AI agents suggest changes to improve the performance or even implement Automatic adjustments based on historical behavior. This proactive approach to monitoring improves the reliability and efficiency of backend systems, reducing downtime and increasing user satisfaction.

V. IMPLEMENTATION PROCESS

To make things simpler and more efficient, our system uses n8n as it a powerful automation tool that connects different apps and works with various aspects of backend components almost from a basic to an advanced working system can be created using this n8n tool, where we can create AI agents without any code very intelligently. This n8n, when combined is capable of working and creating various systems like connecting with frontends, designing intelligent database systems, authentication and security, connecting with LLMs and adapting the environment to work with minimal data and information. It adapts to our instructions through JSON files and responds if they are successful or not and a lot more.

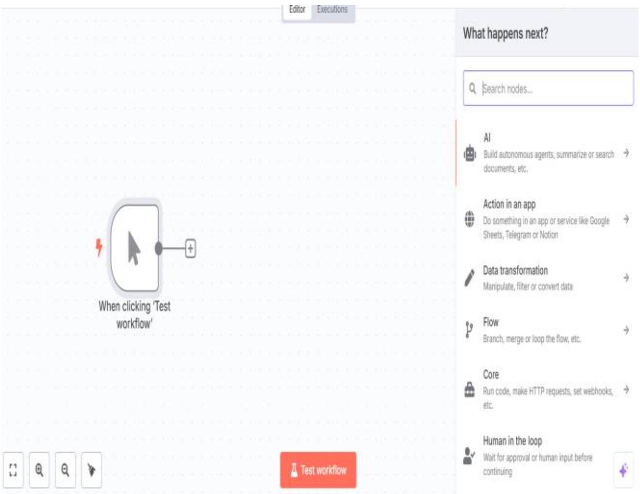
Editor
Executions
When clicking 'Test workflow'
Test workflow
What happens next?
Search nodes...
AI
Build autonomous agents, summarize or search documents, etc.
Action in an app
Do something in an app or service like Google Sheets, Telegram or Notion
Data transformation
Manipulate, filter or convert data
Flow
Branch, merge or loop the flow, etc.
Core
Run code, make HTTP requests, set webhooks, etc.
Human in the loop
Wait for approval or human input before continuing

Fig.1. n8n workflow starting trigger to add nodes

This tool contains various features while creating workflows such as AI agent which is used to build autonomous agents and is capable of connecting various LLMs for conversations, can use basic LLM chains, can extract information from text in a structured format, is capable of answering questions about retrieved documents from database, sentiment analysis, text summarization, text classifier and other AI nodes for embeddings, vector stores etc.

It can also create actions in apps or connect it with various services like Google Sheets, Telegram, Notion. The action in app is used to connect and automate action in the workflows. These are represented in the form of nodes where they are integrated with third party apps or services like Google Sheets, Slack, or AWS or utility like HTTP Request or RabbitMQ. Nodes can send data, retrieve data from or even manipulate data in these services according to the instructions. These nodes are used for various categories such as for cloud providers, CRM and marketing, project management, DevOps and Monitoring, database and storage, communication, security tools, social media and web, APIs and Utils, E-commerce, scheduling, Translation/NLP, Automation & AI and other services.
Few use cases for this can be lead management system, file processing automation, security alert system, social media monitor, e-commerce handling and so on.
Data tranforamtion in this n8n can be used to add code, can edit fields, filter items, limit the items, remove duplicates of matching values, split out the items, combine the items, summarize and to convert the data into files, edit images provide cryptographic utilities, converts binary data to JSON, work with html, markdown, etc.

The flow is used to branch, merge or loop the flow.
The core in this is used to make HTTP requests, setup webhooks, execute data, generate webforms using n8n forms, waits and responds to webhook lastly .

And finally it can trigger the workflows by adding multiple triggers such as trigger manually, on app event, on a schedule, on a webhook call, on a chat message, etc.

From Fig.1 you can see after creating the trigger you can start adding necessary nodes like webhooks, HTTP requests, AI agents, database and end with webhook/HTTP request to receive response.

VI. PROPOSED SYSTEM DESIGN

The proposed system design is as follows:

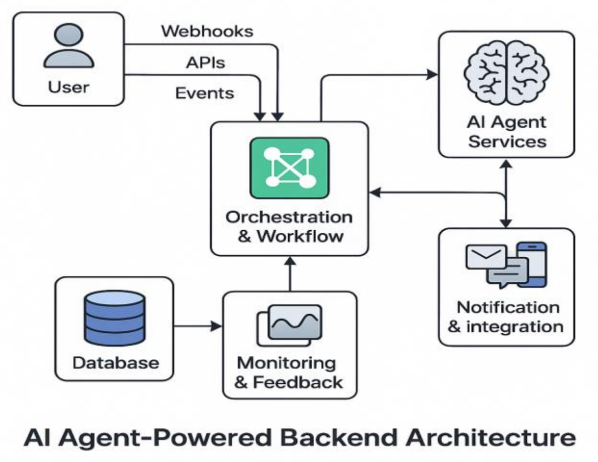

Fig.2. AI agent backend architecture

A. User:

- In this end user interacts with front end app using
- Webhooks - Webhooks are commonly used for event-driven communication of frontend with backend where the frontend triggers the backend workflows in real time and it also helps in posting an event to another system over web using an API.
- Events - Events that are used for user information and cart update

B. Orchestration & Workflow (n8n):

- This is used for core logic builder.
- It receives inputs from front end.
- Then it routes a request through appropriate work flows.

C. AI Agent Services:

- This is the main brain of the system.
- It powers intelligent decisions using LLMS (large language models).
- Functions: It makes decisions using NLP.
- It makes more context aware decisions.
- It dynamically creates data base queries.

D. Notification & Integration Layer:

- Notifications are triggered by AI agents or work loads
- It handles external communications like emails, Voice alerts through phone or Slack etc.
- It also integrates 3 party APIs.

E. Monitoring & Feedback:

- In monitoring & Feedback it uses tools like Splunk, APPD or Dynatrace etc.
- It monitors errors, latencies etc.

F. Database:

- It stores an application's data
- It can be modified or updated by SQL or NO SQL.
- Database agents can dynamically modify queries as it requires.

VII. FRONTEND AND BACKEND INTEGRATION WITH AI AGENTS AND TOOLS LIKE N8N:

In today's application development the coordination between frontend interfaces and backend logic plays a crucial role in delivering a seamless user experience and enhancing application performance. The rise of AI agents particularly when integrated with workflow automation platforms like n8n are reimagining how we think about this integration. Instead of depending upon rigid, hardcoded API endpoints developers are now grasping to a more flexible, declarative and communication models.

This allows for a more modern and interactive architecture. The workflows emphasized by AI agents serve as an adaptable middle layer so that not only process incoming requests but it will also execute complex logic and returns insightful responses as output.

Frontend frameworks such as React, Angular, and Vue.js can now interact directly with n8n workflows by using various methods like HTTP webhooks, REST APIs, web sockets, GraphQL, Server sent events (SSE) [9] and many more. Integrating AI agents with frontend applications can be achieved by various methods each is suited for a specific use case here are a few have a look on them. Webhooks are commonly used for event-driven communication of frontend with backend where the frontend triggers the backend workflows in real time. REST APIs remain as the most traditional approach which enables the request and response interactions between frontend and AI services ideal for sending prompts and receiving appropriate results. For more interactive and live experience we can use WebSockets which offer a persistent and bi-directional connections that allows the real time streaming of data which is perfect for chatbots or live updates like news and many more. GraphQL [10] provides a flexible way to query an AI agent response especially when dealing with complex or nested data structures. Server-Sent Events (SSE) offer a simplified choice to WebSockets for Single-direction streaming which is suitable for running AI tasks. Additionally, many platforms offer SDKs or JavaScript libraries that streamline integration by simplifying API calls making them perfect for quick development. For inserting AI tools or dashboards directly into the frontend you can use iFrames or micro frontends. The choice of the method you choose depends on factors like responsiveness, complexity and how dynamic frontend AI interactions are required to be.

This modern approach keeps the frontend lightweight and agile allowing it to focus on delivering the best user experience. Meanwhile the heavy lifting such as data processing, business logic evaluation and notification management is efficiently handled by AI driven workflows at the backend. This separation concern not only simplifies development but also improves the overall performance and scalability of applications.

VIII. SMART WORKFLOWS WITH AGENTIC AI AND N8N:

n8n is an open-source workflow automation platform that enables the visual connection of services, APIs, and databases without extensive coding requirements. When these workflows are integrated with Agentic AI then this combination creates a powerful automation solution where AI functions will act as a decision maker while n8n executes the necessary actions. The AI component will analyze objectives, will determine appropriate steps and follows the execution plan, n8n then implements these plans by collaborating or interfacing with relevant systems.

This strategic alliance enables intelligent end to end automation across organizational limits. For example, if an agent is given a task to “follow up with new leads” then it can use n8n to fetch data from a CRM (Customer Relationship Management) [11] to send personalized emails, communication and records interaction outcomes without any human involvement.

This type of partnership of AI agents and n8n automation tool allows for intelligent and an end-to-end automation where AI will bring the reasoning and adaptability to the environment and the task and on other hand n8n handles the real-world system interactions effectively.

Such integration significantly simplifies backend operations which include data management, report generation, customer follow up workflows and system maintenance tasks such as system updates significantly simpler and faster. Organizations implementing this combined solution will definitely benefit from reduced manual processing requirements and improved operational efficiency. As a result, enterprises can also achieve smarter and more flexible automations across a wide range of business processes which reduces manual effort and also boosts productivity which in turn saves a lot of time for developers.

IX. CONCLUSION:

This paper presents an innovative approach to backend development which involves integrating AI agents with workflow orchestration platforms like n8n. Our architecture transforms traditional backend development by replacing manual coding by using various natural language driven processes which may include authentication, database management, business logic and other critical components. Our research paper offers a complete solution that addresses the maximum of backend lifecycle instead of separated or independent components.

By modifying application development, our approach enables non-technical stake holders to build more sophisticated systems without much coding experience. This change not only accelerated development but also changes our mind ow we think about backend architecture. As this technology grows, we can visualize how developers are evolving from code writers to system designers who can shape the AI-powered workflows, eventually making software development more accessible while driving innovation across different industries.

REFERENCES

[1] F. Huang, "Data Processing," in Encyclopedia of Big Data, L. Schintler and C. McNeely, Eds. Cham: Springer, 2019. [Online]. Available: https://doi.org/10.1007/978-3-319-32001-4_314-1

[2] M. Land and M. Crnkovic, "Existing Approaches to Software Integration – and a Challenge for the Future," Proceedings of the 2002 International Conference on Software Engineering Research and Practice (SERP), Las Vegas, NV, USA, 2002, pp. 1-7. [Online]. Available: https://www.semanticscholar.org/paper/Existing-Approaches-to-Software-Integration-%E2%80%93-and-a-Land-Crnkovic/60aa78b942b488456e529834ef6b95d9913e7ccb. [Accessed: Apr. 20, 2025].

[3] W. T. Lehto and M. S. Liddell, "A Study on Boilerplate Code in Modern Programming Languages," IEEE Transactions on Software Engineering, vol. 45, no. 6, pp. 845-856, June 2019. doi: 10.1109/TSE.2019.2899047.

[4] P. S. Lee, M. H. Cho, and Y. G. Kim, "Efficient Debugging of Backend Logic in Distributed Systems," IEEE Transactions on Software Engineering, vol. 46, no. 5, pp. 673-688, May 2020. doi: 10.1109/TSE.2020.2990485.

[5] A. J. Smith and L. S. Kumar, "The Role of DevOps in Enhancing Backend System Efficiency," IEEE Transactions on Software Engineering, vol. 45, no. 8, pp. 985-996, August 2020. doi: 10.1109/TSE.2020.2991498.

[6] T. B. Brown, M. Mann, N. Ryder, et al., "Language Models are Few-Shot Learners," Proceedings of NeurIPS, vol. 33, pp. 1877-1901, 2020. doi: 10.5555/3454287.3454752.

[7] B. Ho, "n8n: A Free and Open Workflow Automation Tool," n8n Blog, 2020. [Online]. Available: https://n8n.io/blog/introducing-n8n. [Accessed: Apr. 2025].

[8] J. Smith, "AI Agents in Modern Software Systems: A Comprehensive Review," IEEE Access, vol. 8, pp. 104356-104375, 2020. doi: 10.1109/ACCESS.2020.2994365.

[9] A. Nguyen and J. Lee, "Server-Sent Events for Real-Time Web Applications," IEEE Transactions on Web Engineering, vol. 16, no. 2, pp. 101-113, 2020. doi: 10.1109/TWE.2020.2980165.

[10] A. Williams and B. Patel, "Optimizing API Queries with GraphQL," IEEE Transactions on Software Engineering, vol. 43, no. 10, pp. 1031-1042, 2018. doi: 10.1109/TSE.2018.2798494.

[11] B. Smith and R. Kumar, "Customer Relationship Management: A Strategic Approach to Enhancing Customer Satisfaction," IEEE Transactions on Engineering Management, vol. 56, no. 4, pp. 529-538, 2009. doi: 10.1109/TEM.2009.2028019.