Programming assistance based on Naeural AI OS Platform

Tiberius PÎRCĂLABU[1], Nicolae ȚĂPUȘ[1], Andrei DAMIAN[2]

[1] National University of Science and Technology Politehnica Bucharest, Romania

[2] Naeural SRL

tiberius.pircalabu@upb.ro, ntapus@cs.pub.ro, andrei.damian@naeural.ai

Abstract: In the context of the development of artificial intelligence models and particularly of the large language models (LLM), one area where they showed significant potential is programming assistance, with common tasks frequently encountered in practice, for which generic solutions already exist and can be offered as suggestions by such models. However, a complex infrastructure is required to serve an artificial intelligence model, both in terms of high-performance hardware resources and in terms of the maintenance needed for them and the related software components. This paper proposes the design and implementation of CodeXpand, an online programming assistance service by leveraging the capabilities of the Naeural AI OS, which provides a complete infrastructure for training and serving artificial intelligence models in a decentralized global community based on a token economy. This platform enables the rapid development and deployment of artificial intelligence applications by efficiently allocating infrastructure, reducing costs, and securely distributing tasks across a decentralized global community. The SDK provided by Naeural AI OS for the TypeScript language was used to integrate advanced LLM inferences for implementing a pair of tools assisting with programming tasks: a web-based conversational chat interface around programming topics and a Visual Studio Code custom extension for source code autocomplete suggestions.

Keywords: Coding Assistant, Artificial Intelligence, Large Language Models, Blockchain, Decentralized Economy, Code Completion.

Asistență pentru programare bazată pe platforma Naeural AI OS

Rezumat: În contextul dezvoltării modelelor de inteligență artificială și în special a modelelor mari de limbaj (LLM), un domeniu în care acestea au demonstrat un potențial semnificativ este asistența la programare, cu sarcini comune întâlnite frecvent în practică, pentru care există deja soluții generice care pot fi oferite ca sugestii de astfel de modele. Totuși, acest lucru necesită o infrastructură complexă care să deservească un model de inteligență artificială, atât în ceea ce privește resursele hardware de înaltă performanță, cât și în ceea ce privește întreținerea necesară pentru menținerea acestora și a componentelor software aferente. Prezenta lucrare propune proiectarea și implementarea CodeXpand, un serviciu online de asistență pentru programare prin valorificarea capacităților sistemului de operare Naeural AI OS, care oferă o infrastructură completă pentru formarea și deservirea modelelor de inteligență artificială într-o comunitate globală descentralizată bazată pe o economie a jetoanelor. Această platformă permite dezvoltarea și implementarea rapidă a aplicațiilor de inteligență artificială prin alocarea eficientă a infrastructurii, reducerea costurilor și distribuirea sigură a sarcinilor într-o comunitate globală descentralizată. SDK-ul furnizat de Naeural AI OS pentru limbajul TypeScript a fost utilizat pentru a integra inferențe LLM avansate în scopul dezvoltării unui set de instrumente care ajută la sarcinile de programare: o interfață de chat conversațională bazată pe web în jurul subiectelor de programare și o extensie personalizată Visual Studio Code pentru sugestii de autocompletare a codului sursă.

Cuvinte cheie: Asistent de codare, Inteligență Artificială, Modele Lingvistice Mari, Blockchain, Economie descentralizată, Completarea codului.

1. Introduction

In the context of the rapid development of artificial intelligence, assisting programmers in their daily tasks has become an essential application. Programming assistance not only simplifies the coding process but also improves the efficiency and quality of the produced code. However, to offer such services, a complex infrastructure capable of supporting AI models in production is required. This involves the use of high-performance hardware resources and ensuring constant maintenance of the software infrastructure.

https://doi.org/10.33436/v34i4y202404

Naeural AI OS is an innovative platform that offers a complete infrastructure for training and serving AI models in a decentralized global community, based on a token economy. The platform facilitates the integration of computing technologies into everyday objects and environments, creating a network of interconnected devices that can communicate with each other and with users. This enhanced connectivity allows for more efficient collaboration and a more sustainable community (Voshmgir, 2019; Milik et al., 2023).

Naeural AI OS addresses common problems encountered in the widespread adoption of AI solutions, such as high costs and the complexity of cloud infrastructures. The platform offers a method for rapid development and deployment of AI applications by efficiently allocating infrastructure, reducing costs, and securely distributing tasks in a decentralized global community. This is achieved through the integration of blockchain technologies and the use of microtransactions to ensure fair compensation for the processing nodes.

To realize the potential of blockchain-based platforms in AI democratization, it is essential to rethink traditional business models and embrace the decentralized and cooperation-based paradigms enabled by the blockchain technology. As highlighted in recent research, blockchain facilitates a shift from centralized, isolated business models to decentralized ecosystems where stakeholders across the value chain collaborate to generate value that would otherwise be unattainable individually. This community-oriented approach aligns with the core objectives of Naeural AI OS, promoting resource sharing, cooperation, and the creation of new value-added services (Dehbasteh et al., 2019). The platform's reliance on microtransactions and blockchain-based task distribution further underscores its alignment with such decentralized business model principles, supporting more inclusive and efficient AI-driven ecosystems (Dehbasteh et al., 2019; Milik et al., 2023).

To facilitate the adoption and integration of Naeural AI OS, the platform offers SDKs for multiple programming languages, including Python and TypeScript. These SDKs allow developers to connect to the Naeural network, create and manage end-to-end AI pipelines, and monitor the progress of tasks. In this project, the TypeScript SDK was used to develop an online programming assistance service.

In the subsequent sections, a review of related work on comparable services will be conducted, also providing a description of the Neural AI OS platform and its application in the study, as well as outlining the experimental setup along with the results obtained. Finally, the conclusions will be presented together with the directions for future research.

2. Related work

AI assistance for programming has become a major area of interest, with numerous tools developed to improve the productivity and efficiency of software developers. Among the most notable solutions are GitHub Copilot and Amazon CodeWhisperer. These tools offer various functionalities, from automatic code completion and generation of entire functions to refactoring suggestions and bug detection. Additionally, there are chat-based assistance services based on LLMs, such as ChatGPT, Anthropic, and Phind, which indirectly facilitate assistance for programmers through detailed responses and relevant suggestions.

2.1. Code autocompletion

GitHub Copilot, created by GitHub, OpenAI, and Microsoft, is a widely recognized AI assistant for programming. Powered by OpenAI's GPT-3, it provides contextual code suggestions, automatic completions, and entire functions. Copilot integrates seamlessly with popular IDEs like Visual Studio Code, Visual Studio, and JetBrains, easing its adoption by developers. Notably, it can learn from natural language comments to suggest relevant code, simplifying development. Its integration with GitHub and support for multiple programming languages make it a versatile tool for developers across various fields (Friedman, 2022).

Amazon CodeWhisperer is another AI assistant for programming developed by AWS that supports a wide range of programming languages, including Python, Java, JavaScript, and TypeScript. It is integrated with AWS platforms and can provide real-time code completions and suggestions, enhancing developer productivity. A distinctive feature of CodeWhisperer is its vulnerability scanning function, which helps identify and fix security issues in the code. CodeWhisperer is designed to work well with other AWS services, such as AWS Cloud9, AWS Lambda, and Amazon SageMaker, thereby facilitating the development and deployment of AI applications in the AWS environment. It also offers comment completions and documentation suggestions, contributing to the creation of a better-documented and more maintainable code (AWS Machine Learning Blog, 2022).

2.2. Conversational user interfaces

In addition to source code autocompletion tools, there are also chat-based conversational user interfaces based on large language models that facilitate direct or indirect assistance for programmers.

The first one is ChatGPT, which came out as a notable breakthrough in public engagement with transformer models (OpenAI, 2022). Initially based on the slightly less powerful GPT 3.5 model, it then upgraded to the GPT-4 model to answer complex questions and provide detailed programming suggestions. It can help programmers understand difficult concepts, find solutions to specific problems, and generate code examples (OpenAI et al., 2023). ChatGPT has proven effective in expediting development processes, particularly during the initial stages of coding and debugging. It also demonstrated significant potential for accelerating workflows in areas such as computer vision and multimedia applications (Bolcaș, 2024).

Anthropic is another company that develops advanced language models for various applications, including programming assistance. Their models are trained to be safe and useful, providing precise and contextual suggestions within chat interactions (Anthropic, 2023).

Phind is a search engine optimized for developers, using large language models to provide quick and relevant answers to technical questions. It can help programmers quickly find solutions and code examples from reliable sources (Royzen, 2024).

These tools reflect the diversity and innovation in the field of AI assistance for programming, each offering various features to improve developers' workflows and their specific needs. Our work integrates into this context, using the Naeural AI OS platform to develop a programming assistance service with the TypeScript SDK, both through a conversational user interface and a custom extension for an integrated development environment.

3. Naeural AI OS Platform

Widespread adoption of AI is often difficult because of high costs and the complexity of necessary infrastructures. Naeural AI OS reduces these barriers by utilizing a decentralized network of devices that share computing resources. It ensures a high level of security and privacy by using blockchain technologies for secure and immutable transactions, as well as private consensus mechanisms for data validation and aggregation. The platform is designed to be scalable and flexible, allowing integration with various data sources and IoT devices. This facilitates the rapid development of AI applications through a modular architecture and a configurable execution engine.

The architecture of Naeural AI OS illustrated in Figure 1 (Milik et al., 2023) is composed of several essential components that ensure the efficient and secure operation of the platform:

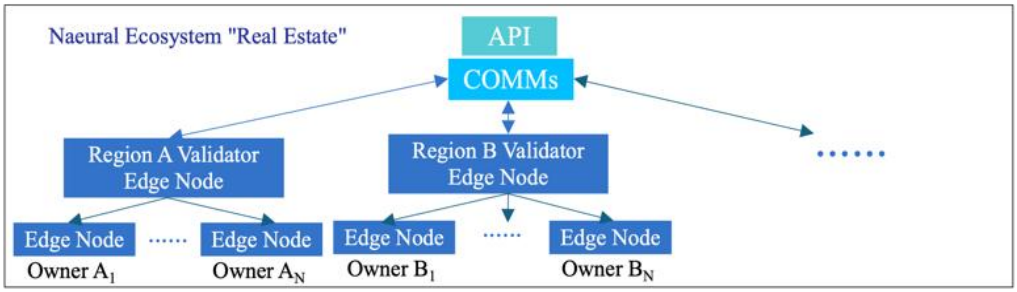

Figure 1. Naeural AI OS Architecture

The heart of the platform, E2 (Naeural Execution Engine), is responsible for managing and executing AI tasks. It is compatible with a wide range of operating systems and hardware devices, including those with advanced GPU computing capabilities. The platform uses a decentralized mechanism for distributing processing tasks among participating nodes. This system is supported by a private blockchain for internal transactions and a public blockchain for microtransactions, thus ensuring fair compensation for processing nodes.

To facilitate platform adoption, Naeural AI OS offers SDKs for multiple programming languages, including Python and NodeJS. These SDKs allow developers to connect to the network, create and manage AI pipelines, monitor task progress and get back AI model inference results.The platform uses a token-based economy to motivate and reward users who contribute computing resources. Each processing node is identified and recognized in the network through a non-fungible utility token on the public blockchain.

4. Experimental setup

4.1. Specifications

The first step in building the application that offers the online programming assistance service was studying the working patterns provided by similar services. Two of the most commonly used are chat-based conversations with a virtual assistant and code autocomplete suggestions in integrated development environments.

Based on the commonly used functions for "Software as a Service", a list of software features was constructed around the two programming assistance patterns to ensure the developed application meets the requirements for offering these services on a monthly subscription model. The features were described using "user stories", a common practice in Agile development methodology, which represents a clear and easily understandable perspective from end-users. These methods are an integral part of the user-centered design process, ensuring that development aligns with the real needs and expectations of the users (Norman & Draper, 1986; Beck, 2000; Beck, 2001; Turner, Budgen & Brereton, 2003).

The main features of CodeXpand include the ability for users to access informative content such as a landing page and pricing information, account management functionalities like registration, login, subscription management, and profile updates, and core application features related to programming assistance. The core features are: initiating chat threads for coding help, receiving and interacting with programming advice, syntax-highlighted code suggestions, revisiting and searching past interactions, as well as the integration of a custom extension for Visual Studio Code that offers autocomplete suggestions tailored to the user's coding context.

4.2. Architecture

The architecture for the CodeXpand service, as depicted in Figure 2, was designed in alignment with the list of software features outlined through user stories.

The chat-based conversations can be conducted from the user's browser through the chat interface directly provided from a web application. For storage, the web application stores

everything into a local database. The second tool, which offers source code autocomplete suggestions, is a standalone application implemented as an extension for Visual Studio Code, labelled as "VS Code Extension" in Figure 2. It communicates with the web application via a public REST API, which the application exposes.

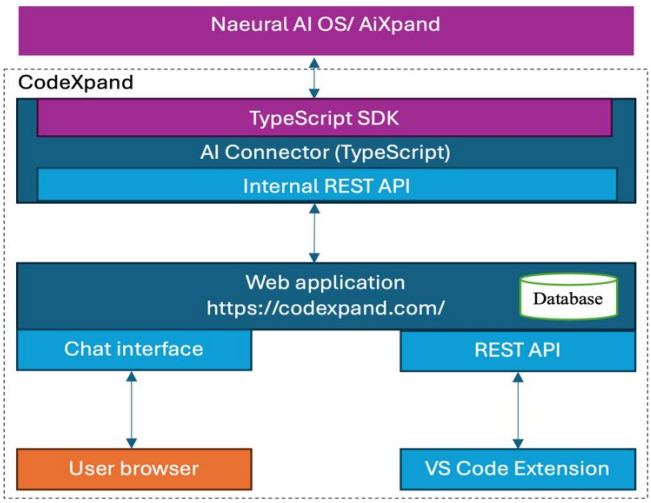

Figure 2. CodeXpand Architecture

The communication between CodeXpand and the Naeural AI OS platform is mediated using the "AI Connector" component, a separate application that was created to centralize the business logic related to accessing the external AI services, including platform specific functions implemented by using the TypeScript SDK. This approach allows for future maintenance or updates to platform specifications to be managed in a single location and facilitates potential interfacing with other platforms offering similar services. The functions of the "AI Connector" application are provided exclusively to the web application through a private REST API called "Internal REST API", with both applications running on the same server and implementing security measures to prevent unauthorized access. The web application offers a browser accessible chat interface for conversations with the user and a public REST API for communicating with the "VS Code Extension" that offers in context autocomplete suggestions directly inside Visual Studio Code source code editing window.

4.3. Data Flows

Depending on the tool used at a given point, the data flows are depicted in Figure 3:

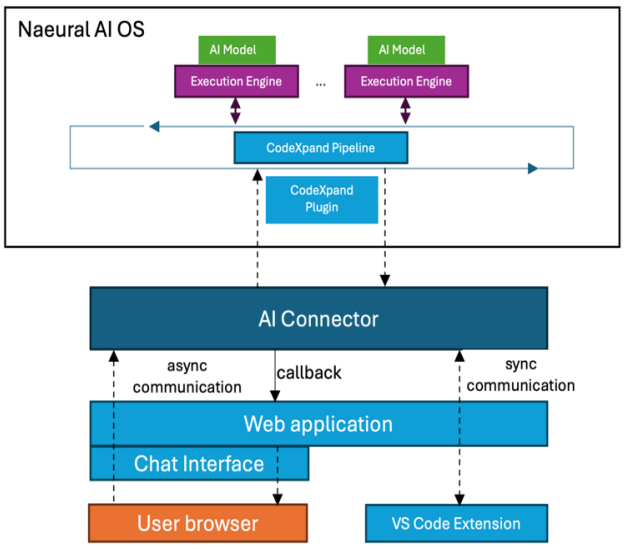

Figure 3. Information exchange between CodeXpand and Naeural AI OS

4.3.1. Chat interface

The user interacts with the web application through the browser for the chat service. When a new conversation is initiated, the text message is sent by the web application to the "AI Connector", which in turn sends the message to the Naeural AI OS platform, along with a callback URL address hosted on the web application, which can later be called to post the response. The connection between the web application and the "AI Connector" is closed immediately. The Naeural AI OS platform uses an AI model to formulate a response and returns it to the "AI Connector" component, which asynchronously posts the response to the web application by calling the callback URL address received with the initial message. In this way, the user's conversation with the AI model can be completed with the AI's response even if the user closes the browser and returns later.

If the user is still present in the chat interface when the response is received, it will be visible almost instantly, as the frontend chat component periodically polls to check if the conversation has been updated.

4.3.2. Extension for Visual Studio Code

For source code autocomplete suggestions in the Visual Studio Code environment, communication with the web application is done synchronously, waiting for a response after each request sent, unlike in chat discussions. Thus, within a source file opened in Visual Studio Code in one of the languages TypeScript, Python or PHP, if the user has already written part of the source code (e.g. "def fibo (" in Python) and pauses for 2 seconds, the extension sends a request to the public REST API provided by the web application. This API is responsible for authenticating the right to access CodeXpand services using an API key associated with the user, which can be obtained from the web application while logged into their account. In addition to the authentication key, the request includes the source code written up to the cursor position. After correct authentication through the REST API, the web application resends the request to the "AI Connector" component, which forwards it to the Naeural AI OS platform. The connections between the VS Code Extension and the web application and between the web application and the "AI Connector" remain open until a response is received from Naeural AI OS, which is then displayed as a suggestion within Visual Studio Code source code editing window.

4.4. Implementation

The experiment aimed to prove that a functional code assistant can be built on top of decentralized services provided by the Neural OS AI platform. In order to achieve this purpose, the implementation phase of CodeXpand focused on the development of three key elements: a web application for user interaction and management, an AI connector to interface with the Naeural AI OS platform, and an extension for Visual Studio Code to provide real-time code suggestions.

For implementing the web application, PHP version 8.x with the Laravel framework version 10.x was selected for the back end, MySQL version 8.x as the database engine, and the VueJS framework version 3.x for the front-end in the application's tech stack.

As the Naeural AI OS platform does not offer a direct SDK for the PHP language, the "AI Connector" component was implemented in TypeScript, for which an SDK is provided. This component serves as an intermediary between CodeXpand and the Naeural AI OS platform. It runs continuously in a loop as a server process, capable of receiving commands via the private "Internal REST API" from the web application, forwarding them to Naeural AI OS, and then returning the responses back to the initiator.

In the "AI Connector" component, specific concepts of the Naeural AI OS platform were used for integration with its services. Thus, the component runs as a background service, using a supervisor. It initially connects to the platform's MQTT server and authenticates in a first step based on a set of credentials, including a private key for authentication on the platform's

blockchain. In this way, the CodeXpand application integrates into the decentralized community of Naeural AI OS as a user of Large Language Model (LLM) AI services. After successful authentication, the component connects to a dedicated pipeline for CodeXpand services, ensuring that an instance of a customized plugin with a predefined signature is attached.

The pipeline concept used in the Naeural AI OS platform allows for the definition of a continuous data flow, to which command information can be sent by certain entities, in our case, the CodeXpand application, through the "AI Connector" component, while other, internal entities can retrieve the command information, execute them on distributed nodes, and then return the execution results in the same data flow. The entity that sends the commands can identify results related to a command through a customized plugin that defines a signature accompanying the command information and subsequently the execution results. Through the TypeScript SDK, the "AI Connector" component receives an event when information containing this signature appears and retrieves the useful data, which it then forwards to the web application.

In order to receive commands from the web application, the "AI Connector" component exposes three API endpoints:

- */chat (POST)*: for chat discussion messages. Messages received on this endpoint include a web callback address for asynchronous responses. Message details are added as a "chat" command in the Naeural AI OS pipeline, along with callback metadata. The SDK filters responses from the AI model based on the plugin signature, identifies the appropriate callback from the metadata, and sends the response;
- */autocomplete (POST)*: for messages related to autocomplete requests, with the message type "autocomplete," for whom details are sent to the pipeline as commands. Unlike the chat endpoint, no callback is used; the connection remains open until a response to the initial request is identified;
- */systemHealth (POST)*: allows the web application to check the connection status with the Naeural AI OS platform. For example, this endpoint will return the text "Down" if the active pipeline or the custom plugin attached to it is not found.

The extension for Visual Studio Code was developed in TypeScript and interfaces with the same intermediary component for communication with Naeural AI OS. Although a direct integration with the TypeScript SDK could have facilitated direct communication with Naeural AI OS, this solution was chosen for two reasons: first, to authenticate the user using the extension with their API key, accessible from their CodeXpand account in the web application, and second, to centralize the business logic for communication with the Naeural AI OS platform within the "AI Connector" component, which is closely coupled with the TypeScript SDK.

5. Results

The final application was installed on the domain codexpand.com as a proof of concept. It allowed signups using an invite code. A $1 subscription was required to be able to use the service, once a user had successfully created an account. Because it was only a proof of concept, the Stripe integration was set to sandbox mode, which allowed using test credit card information for activating the subscription.

The initial dashboard shows the system status and links to the main tools. For chatting, specific help can be requested for a task in a designated programming language, as seen in Figure 4.

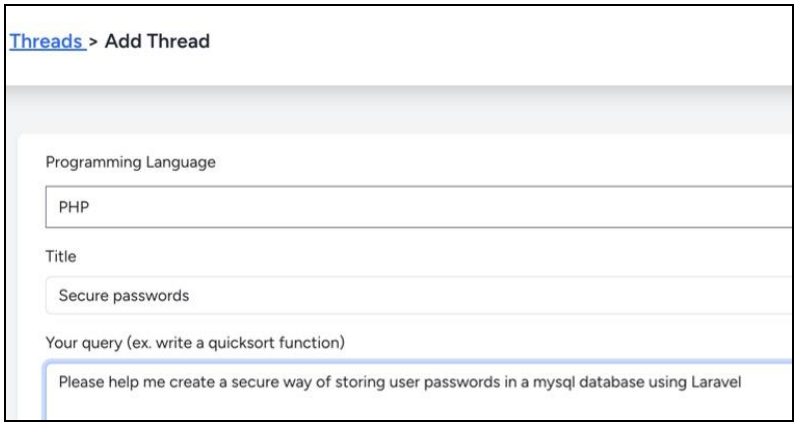

Figure 4. Sending a request to the chat service

Figure 5 shows the answer coming back asynchronously and includes both explanations and code ready to be copied and pasted elsewhere.

Figure 5. Receiving an answer from the chat service

The autocomplete extension can be downloaded from a dedicated page, where instructions help the user install, configure and activate it, as illustrated in Figure 6.

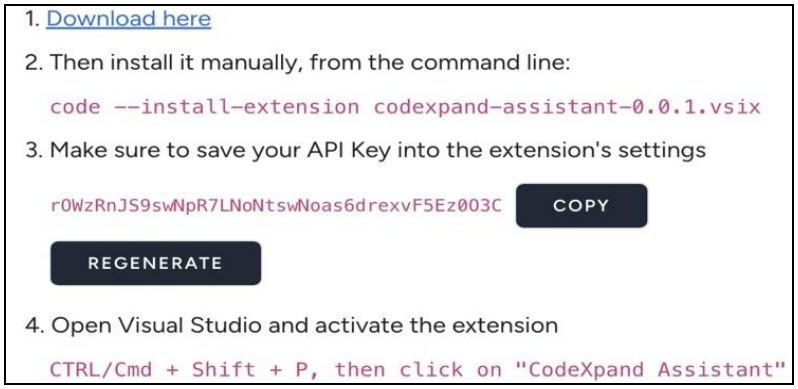

Figure 6. Download and installation instructions for the Visual Studio Code extension

After successful installation and activation, given the user has opened a source code file for one of the supported languages, e.g. Python, once a few letters have been typed and a 2 seconds pause is detected, the extension sends the autocomplete request to CodeXpand, including the entire source code until the current cursor position.

This request is synchronous, therefore in a few seconds the answer is received and the autocomplete suggestion is displayed in a different color, with the user having the option to choose it by pressing the [Tab] key or ignore it by pressing the [Esc] key, as seen in Figure 7.

```
def fibo(m):
    if m <= 1:
        return m
    else:
        return fibo(m-1) + fibo(m-2)
```

Figure 7. Extension for Visual Studio Code offering autocomplete suggestion

6. Conclusions and future work

This project aimed to demonstrate the practical application of Naeural AI OS by providing programming assistance through AI chat interactions and autocomplete suggestions within integrated development environments for multiple programming languages. By utilizing the TypeScript SDK, key concepts and features of the Naeural AI OS were successfully integrated. The applications were deployed on a publicly accessible website and tested in real-world scenarios, including challenges encountered during the development of this application. The chosen pair of tools validated both asynchronous and synchronous communication methods with the Naeural AI OS platform, underscoring its versatility for various real-world applications.

Due to legal implications associated with a paid subscription-based official application, this project is currently implemented as a proof of concept and is not yet widely available to the public. Consequently, certain parameters such as performance and scalability have not been extensively tested or optimized. The application is hosted on a server with modest hardware specifications, limiting its suitability for handling higher loads. Currently, the application supports only three programming languages: Python, PHP, and TypeScript. Moreover, the autocomplete extension is implemented exclusively for the Visual Studio Code IDE. The user interface was designed with a focus on efficiency and delivering the project's value proposition quickly, thus aesthetic considerations were minimized.

Further research is needed to evaluate the viability of this project on a larger scale, involving more users and increased load. Upon resolving the legal implications of publicly releasing this project, an optimized configuration for the hosting server will be necessary to facilitate scalability for higher loads. Expanding support for additional programming languages should be straightforward, given that the underlying AI model supported by the Naeural AI OS platform is compatible. Additionally, developing plugins or extensions for other popular IDEs such as JetBrains IDEs, Eclipse, and others would enhance the project's utility and accessibility.

In terms of the token economy, once the project achieves a level of profitability, a smart contract can be established between CodeXpand and the underlying blockchain layer of Naeural AI OS. This will ensure that the services provided by the platform are compensated fairly and transparently.

While the immediate user-facing features of CodeXpand may appear similar to those of existing tools, the underlying use of Naeural AI OS introduces several key advantages. First, the decentralized nature of the platform enhances the system's scalability and resilience, as it does not depend on a single cloud provider. Second, the blockchain-based economy provides a transparent and fair compensation model for resource contributors, which is a significant departure from the centralized billing models of traditional cloud services.

The integration of Naeural AI OS into CodeXpand serves as a proof of concept for how decentralized AI platforms can be used to power real-time, interactive applications. This represents a novel contribution to the field of AI-driven programming assistance, highlighting the potential of decentralized infrastructures to support complex AI tasks in a way that is both economically sustainable and technically robust.

In conclusion, while existing AI programming assistants like GitHub Copilot and Amazon CodeWhisperer have set a high standard for developer tools, they are constrained by their reliance on centralized infrastructure. The introduction of Naeural AI OS as the AI inference provider in CodeXpand showcases the viability of decentralized AI platforms for delivering similar

functionalities with added benefits in terms of scalability, security, and economic fairness. This work not only demonstrates the practical application of Naeural AI OS but also opens up new avenues for research into decentralized AI systems and their potential to transform software development.

Acknowledgement

This paper is a collaboration between POLITEHNICA Bucharest and KNOWLEDGE INVESTMENT GROUP SRL in CloudPrecis POC SIMS 124812 and ReDeN SMIS 156084 projects.

REFERENCES

Anthropic (2023) *Introducing Claude Anthropic.* https://www.anthropic.com/news/introducing-claude [Accessed 11th November 2024].

AWS Machine Learning Blog (2022) *Introducing Amazon CodeWhisperer, the ML-powered coding companion.* https://aws.amazon.com/blogs/machine-learning/introducing-amazon-codewhisperer-the-ml-powered-coding-companion/ [Accessed 11th November 2024].

Beck, K. (2000) *Extreme programming explained: Embrace Change*. Addison-Wesley Professional.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J. et al. (2001) *Manifesto for Agile Software Development*. https://agilemanifesto.org/iso/en/manifesto.html [Accessed 11th November 2024].

Bolcaș, R.-D. (2024) Generating FER models using ChatGPT. *Romanian Journal of Information Technology and Automatic Control [Revista Română de Informatică și Automatică]*. 34(2), 85-96. doi:10.33436/v34i2y202407.

Dehbasteh, K., Pourebrahimi, A., Valmohammadi, C., Afshar Kazemi, M. A. (2019) Identification of the determinants of Blockchain-based business model using hybrid method: Content analysis & System Dynamics. *Romanian Journal of Information Technology and Automatic Control [Revista Română de Informatică și Automatică]*. 29(4), 17-34. doi:10.33436/v29i4y201902.

Friedman, N. (2022) *Introducing GitHub Copilot: your AI pair programmer*. https://github.blog/news-insights/product-news/introducing-github-copilot-ai-pair-programmer/ [Accessed 11th November 2024].

Milik, B., Saraev, S., Bleotiu, C., Lupaescu, R., Hobeanu, B., Damian, A.I. (2023) *Naeural AI OS -- Decentralized ubiquitous computing MLOps execution engine.* [Preprint] https://arxiv.org/abs/2306.08708 [Accessed 11th November 2024].

Norman, D. A. & Draper, S.W. (1986) *User centered system design: New Perspectives on Human-computer Interaction.* Hillsdale, N.J. : L. Erlbaum Associates.

OpenAI (2022) *Introducing ChatGPT.* https://openai.com/index/chatgpt/ [Accessed 11th November 2024].

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L. et al. (2023) *GPT-4 Technical Report.* [Preprint] https://arxiv.org/abs/2303.08774v6 [Accessed 11th November 2024].

Royzen, M. (2024) *Introducing Phind-70B – closing the code quality gap with GPT-4 Turbo while running 4x faster*. https://www.phind.com/blog/introducing-phind-70b [Accessed 11th November 2024].

Turner, M., Budgen, D. & Brereton, P. (2003) Turning software into a service. *Computer*. 36, 38–44. doi: 10.1109/MC.2003.1236470

Voshmgir, S. (2019) *Token economy: How Blockchains and Smart Contracts Revolutionize the Economy*. Illustrated ed. BlockchainHub.

Tiberius PÎRCĂLABU is a doctoral candidate in Artificial Intelligence at the National University of Science and Technology Politehnica Bucharest, Romania, where he previously completed a Master's Degree in Computer Science. He is the founder of a software development firm that delivers advanced technical solutions to clients both within Romania and internationally. His professional expertise spans web applications, desktop applications, and technological stacks tailored for startup environments. His research interests include artificial intelligence, software engineering, and the development of sophisticated algorithms to address real life practical challenges.

Tiberius PÎRCĂLABU este doctorand în Inteligență Artificială la Universitatea Națională de Știință și Tehnologie Politehnica București, unde a absolvit anterior un masterat în Informatică. Este fondatorul unei firme de dezvoltare software care oferă soluții tehnice avansate clienților din România și din străinătate. Expertiza sa profesională acoperă aplicații web, aplicații desktop și pachete tehnologice adaptate pentru medii de start-up. Interesele sale de cercetare includ inteligența artificială, ingineria software și dezvoltarea de algoritmi sofisticați pentru a răspunde provocărilor practice din viața reală.

Nicolae ȚĂPUȘ is a Professor of Computer Science and Engineering, National University of Science and Technology Politehnica Bucharest, Romania. His current research interests include computer architecture, networking, distributed systems, network of sensors, embedded systems.

Nicolae ȚĂPUȘ este profesor de informatică și inginerie, Universitatea Națională de Știință și Tehnologie Politehnica București, România. Interesele sale actuale de cercetare includ arhitectura calculatoarelor, rețele, sisteme distribuite, rețele de senzori, sisteme integrate.

Andrei DAMIAN, Ph.D., is a university lecturer and data scientist with a strong commitment to advancing the accessibility of artificial intelligence (AI) and blockchain technologies. His professional focus lies in bridging the gap between AI and real-world applications. In 2024 he co-founded Naeural SRL, a private company dedicated to the development and implementation of the Naeural AI OS platform within the business sector.

Andrei DAMIAN este lector universitar și cercetător în știinta datelor, dedicat promovării accesibilității inteligenței artificiale (AI) și tehnologiilor blockchain. Activitatea sa profesională se concentrează pe reducerea decalajului dintre AI și aplicațiile din lumea reală. În 2024 a co-fondat Naeural SRL, o companie privată dedicată dezvoltării și implementării platformei Naeural AI OS în sectorul de afaceri.