DOI: https://doi.org/10.36910/6775-2524-0560-2026-62-09
UDC 004.4
Povstiana Yuliia, Ph.D., Associate Professor
https://orcid.org/0000-0001-5426-4157
Lishchyna Nataliia, Ph.D., Associate Professor
https://orcid.org/0000-0002-5200-536X
Surynovych Olena, Ph.D., Associate Professor
https://orcid.org/0000-0002-9300-0039
Boiko Lev, Ph.D.,
https://orcid.org/0009-0001-0117-8551
Kachula Ivan, Master's Student
https://orcid.org/0009-0002-8483-9315
Lutsk National Technical University, Lutsk, Ukraine

ARCHITECTURAL AND ENGINEERING ASPECTS OF INTEGRATING THE NOVITA AI API INTO A WEB APPLICATION FOR IMAGE GENERATION

Povstiana Y., Lishchyna N., Surynovych O., Boiko L., Kachula I. Architectural And Engineering Aspects Of Integrating The Novita Ai Api Into A Web Application For Image Generation. This paper investigates architectural and software engineering approaches to integrating the Novita AI image generation service into a web-based application that supports scalable and asynchronous request processing. The proposed solution is based on a clear separation of responsibilities between frontend and backend components, where the backend handles validation, orchestration of long-running tasks, interaction with external APIs, and data persistence. Communication is implemented using REST for synchronous operations and WebSocket-based notifications for asynchronous updates, while resource-intensive tasks are executed through a queue-based mechanism to prevent interface blocking. Integration with the Novita AI service is encapsulated within a dedicated service layer, ensuring modularity, maintainability, and extensibility. An experimental evaluation was conducted to measure execution times of key operations, including image generation, transformation, upscaling, and custom model training. The results confirm the effectiveness of asynchronous processing and demonstrate that custom models improve output consistency at the cost of a minor increase in generation time. The novelty of this study lies in the architectural justification of asynchronous API integration for AI-based image generation with support for custom model training, validated through experimental performance evaluation.
Keywords: web application, software architecture, software engineering, generative models, API integration, asynchronous processing.

Повстяна Ю.С., Ліщина Н.М., Суринович О.М., Бойко Л.С., Качула І.М. Архітектурні та інженерні аспекти інтеграції API Novita AI у вебзастосунок для генерації зображень. У статті досліджуються архітектурні та інженерні підходи до інтеграції сервісу генерації зображень Novita AI у вебзастосунок, що підтримує масштабовану та асинхронну обробку запитів. Запропоноване рішення ґрунтується на чіткому розмежуванні відповідальностей між frontend- та backend-компонентами, де backend забезпечує валідацію, оркестрацію довготривалих завдань, взаємодію із зовнішніми API та збереження даних. Обмін даними реалізовано з використанням REST для синхронних операцій і WebSocket-сповіщень для асинхронних оновлень, тоді як ресурсоємні завдання виконуються через черговий механізм, що запобігає блокуванню інтерфейсу. Інтеграція із сервісом Novita AI інкапсульована в окремому сервісному шарі, що забезпечує модульність, підтримуваність і розширюваність системи. Проведено експериментальне оцінювання часу виконання ключових операцій, зокрема генерації зображень, трансформації, апскейлінгу та навчання користувацьких моделей. Отримані результати підтверджують ефективність асинхронної обробки та демонструють, що використання користувацьких моделей підвищує узгодженість результатів за рахунок незначного збільшення часу генерації. Наукова новизна даного дослідження полягає в архітектурному обґрунтуванні асинхронної інтеграції API для генерації зображень на основі штучного інтелекту з підтримкою навчання користувацьких моделей, підтвердженому експериментальною оцінкою продуктивності.
Ключові слова: веб застосунок, програмна архітектура, програмна інженерія, генеративні моделі, інтеграція API, асинхронна обробка.

Statement of the Scientific Problem. The rapid development of artificial intelligence technologies has led to the widespread adoption of generative models in the creation of digital visual content. Such models are extensively applied in design, marketing, e-commerce, education, and software development, where there is a growing demand for fast and automated image generation based on textual descriptions. However, standard generative models often fail to fully meet specific business requirements, particularly in terms of accurately reproducing brand identity, corporate visual style, or highly specialized objects.

In this context, an important scientific and practical challenge is the integration of image generation services into web applications with support for training and managing custom models. Of particular interest are platforms that provide API access to generative models and support asynchronous request processing, scalability, and reproducibility of results. This issue is directly related to the practical tasks of digital business transformation and the automation of visual content creation.

Analysis of Research and Publications. Study [1] presents a comprehensive survey of generative diffusion models as a distinct class of deep generative algorithms, addressing their theoretical and mathematical foundations, algorithmic modifications proposed in the literature, and the main directions of practical application, particularly in image synthesis tasks.

The current state of research in the field of image generation is reflected in survey studies focused on the comparative analysis of text-to-image and image-to-image generation approaches. In particular, [2] provides a systematic review of the principal architectures of generative models, including variational autoencoders (VAEs), generative adversarial networks (GANs), and diffusion models, and analyzes their advantages, limitations, and suitability for various application scenarios. Special attention is given to the problem of preserving semantic and structural consistency of generated images, which is especially important in tasks involving scientifically meaningful visual content.

Several applied works have proposed high-performance algorithms and software for analytical processing and recognition of complex structured images [3]. Specifically, approaches combining efficient contour extraction and raster processing demonstrate practical feasibility for real-time image analysis in applied systems, highlighting the importance of algorithmic choices in overall system performance.

Comparative reviews of practical image generation services indicate that existing platforms differ not only in synthesis quality, but also in terms of controllability, ease of integration into application systems, support for custom models, and infrastructure scalability [4]. In this context, particular interest is directed toward platforms that combine ready-to-use generative models with the capability for further training and programmatic integration through standardized interfaces.

In technical documentation and overview materials [4-6], Novita AI is considered as an example of such a platform, as it integrates a library of generative models, a GPU-based serverless infrastructure, and support for asynchronous request processing via an API. These characteristics make such platforms suitable for integration into scalable web applications with high request throughput. These sources are used to describe the implementation context rather than as theoretical foundations.

Recent research has increasingly focused on architectural principles for deploying and integrating machine learning services within cloud-native environments. In particular, the authors of [7] analyze design approaches for cloud-native AI systems, emphasizing scalability, modularity, resource orchestration, and performance trade-offs in AI service deployment. Their findings highlight the importance of service decomposition, containerization, and asynchronous processing when integrating AI-driven functionalities into web-oriented systems.

In contrast to predominantly theoretical studies, a separate group of works focuses on combining the analysis of generative models with the investigation of engineering aspects of their integration into web-based applications. In particular, one study [8] presents a structured overview of image-to-image generative architectures, while another work [9] proposes a methodology for evaluating web API performance based on response time, stability, and throughput, enabling the identification of architectural bottlenecks and the justification of integration decisions.

Thus, existing research covers both the theoretical foundations of generative models and selected aspects of their application in service-oriented solutions. However, despite the substantial body of work devoted to generative models and platforms providing such services, issues related to the practical integration of these platforms into web applications from the perspectives of software architecture and asynchronous processing remain insufficiently systematized, which determines the relevance of the present study.

Purpose of the Study. The purpose of this paper is to investigate approaches to integrating the Novita AI service into a web application for image generation and custom model training, as well as to evaluate the effectiveness of this approach from the perspective of practical application.

Presentation of the Main Material and Justification of the Research Results. To implement the web application, a modern technology stack was employed, oriented toward scalability, maintainability, and integration with external artificial intelligence services. The server-side component was developed using the Laravel framework, which provides support for the MVC architecture, convenient interaction with REST APIs, task queue mechanisms, and database integration. The client-side component was implemented using Vue.js in combination with the Nuxt framework, enabling a component-based user interface architecture, server-side rendering, and efficient routing. Data persistence is ensured through the use of PostgreSQL as a relational database management system with support for transactions and complex queries. Asynchronous processing is implemented using Horizon queues, while event-driven communication between the server and the client is realized through WebSocket connections (Laravel Reverb). Containerization and environment reproducibility are ensured through Docker, using the Laradock setup.

The system architecture is designed according to the principles of service decomposition and event-driven interaction. The frontend is responsible for request formation and result visualization, whereas the backend performs validation, process orchestration, and interaction with the external Novita AI API, while computationally intensive operations are executed on the external service side. Such a distribution of responsibilities minimizes coupling between components and enhances system scalability.

The interaction between system components is formalized in the form of a sequence diagram illustrating the image generation process (Fig. 1).

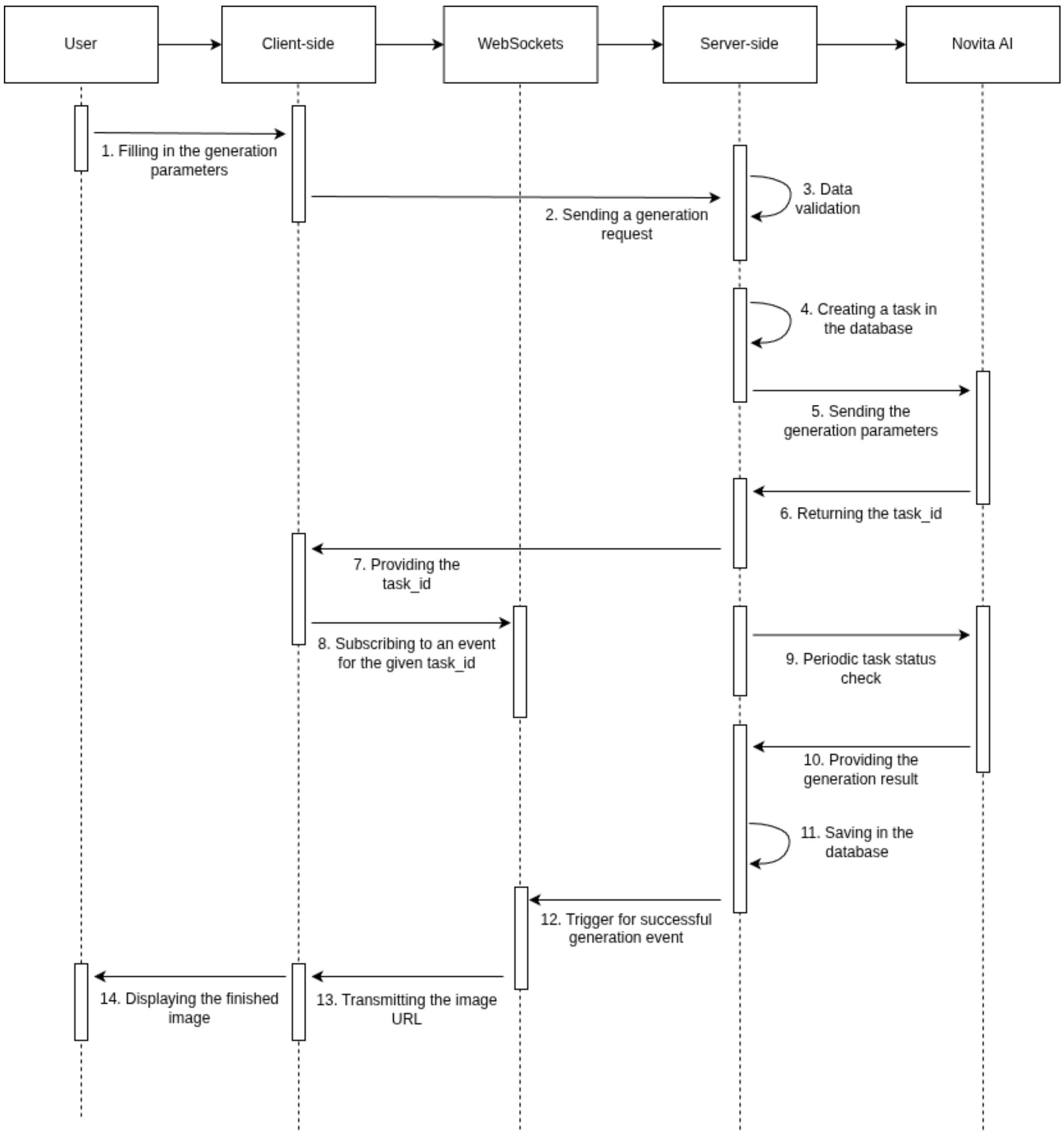
User
Client-side
WebSockets
Server-side
Novita AI
1. Filling in the generation parameters
2. Sending a generation request
3. Data validation
4. Creating a task in the database
5. Sending the generation parameters
6. Returning the task_id
7. Providing the task_id
8. Subscribing to an event for the given task_id
9. Periodic task status check
10. Providing the generation result
11. Saving in the database
12. Trigger for successful generation event
13. Transmitting the image URL
14. Displaying the finished image

Fig. 1. Sequence diagram of the image generation process

The diagram illustrates the complete lifecycle of a request, including user initiation, server-side validation, creation of an asynchronous task, transmission of the task to the Novita AI API, reception of the result, and event-based notification of the client upon completion. The use of an asynchronous processing model prevents user interface blocking and ensures stable system operation under concurrent execution of a large number of requests.

To experimentally validate the system's functionality, a series of tests was conducted with the execution time of the core operations being recorded. The study covered text-to-image generation, image-to-image generation, background and text removal, image resolution upscaling, and custom model training. All experiments were performed in a controlled environment with fixed software versions and a stable network connection to minimize the influence of external factors. Each operation was executed multiple times, and average execution times were calculated. Execution time was selected as the primary evaluation metric due to its direct impact on user experience and system scalability in web-based AI applications.

The measurement results demonstrated that image generation and upscaling operations require substantially longer execution times compared to editing operations, while custom model training is the most resource-intensive process. This confirmed the appropriateness of employing asynchronous processing for computationally heavy operations and synchronous execution for lightweight tasks. At the same time, the system exhibited stable behavior, as the differences between the minimum and maximum execution times remained within an acceptable range.

Table 1. Execution Time of Operations

Operation	Execution Time, mm:ss.ms		
	Minimum	Maximum	Average
Text-to-image	00:44.64	01:00.89	00:52.36
Image-to-image	00:49.32	01:13.97	00:54.06
Background removal	00:04.60	00:08.02	00:07.53
Text removal	00:3.56	00:7.35	00:04.74
Upscale	00:13.33	00:16.90	00:16.24
Custom model training	48:32.68	58:51.27	54:38.17

Particular attention was devoted to analyzing the impact of using custom models on generation quality and execution time. For this purpose, a comparison was conducted between the results produced by a standard model and those generated by a custom model trained on target-specific data (Fig. 2).

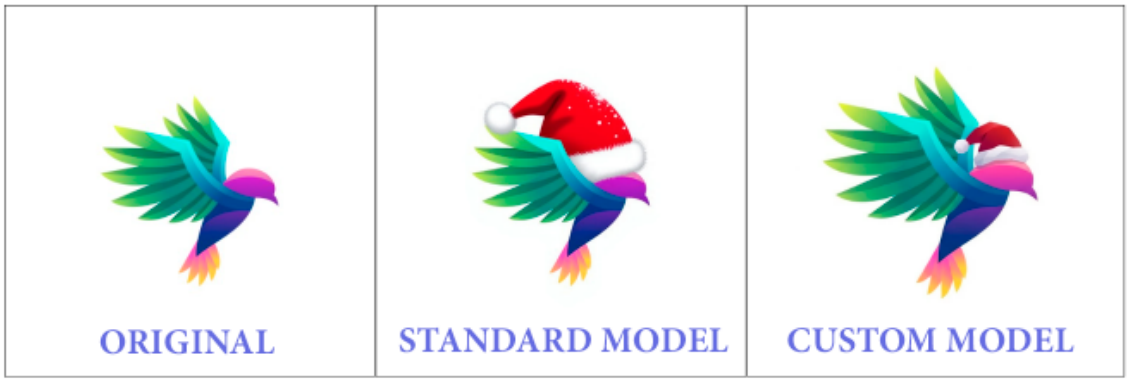

Fig. 2. Comparison of models with identical settings

Figure 3 presents the results obtained under conditions of detailed tuning of the standard model. Although the output quality improves, it still remains inferior to that of the custom model in terms of style reproduction accuracy and preservation of proportions.

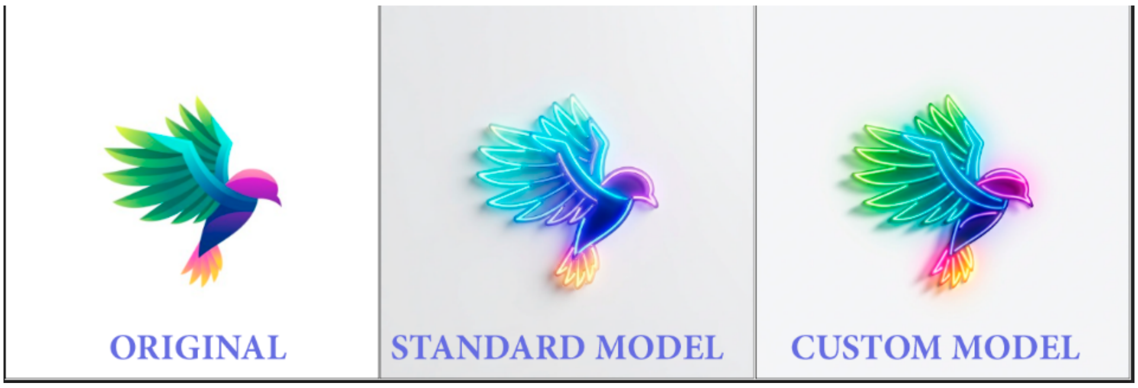

Fig. 3. Comparison of models with different settings

Experimental results indicate that the use of a custom model increases the average image generation time by approximately 5%; however, it significantly reduces the number of iterations required for manual parameter tuning, which overall shortens the total time needed to obtain an acceptable result (Fig. 4). Consequently, custom models improve the system's efficiency in the long term, particularly in scenarios involving repetitive tasks with fixed stylistic or branding requirements.

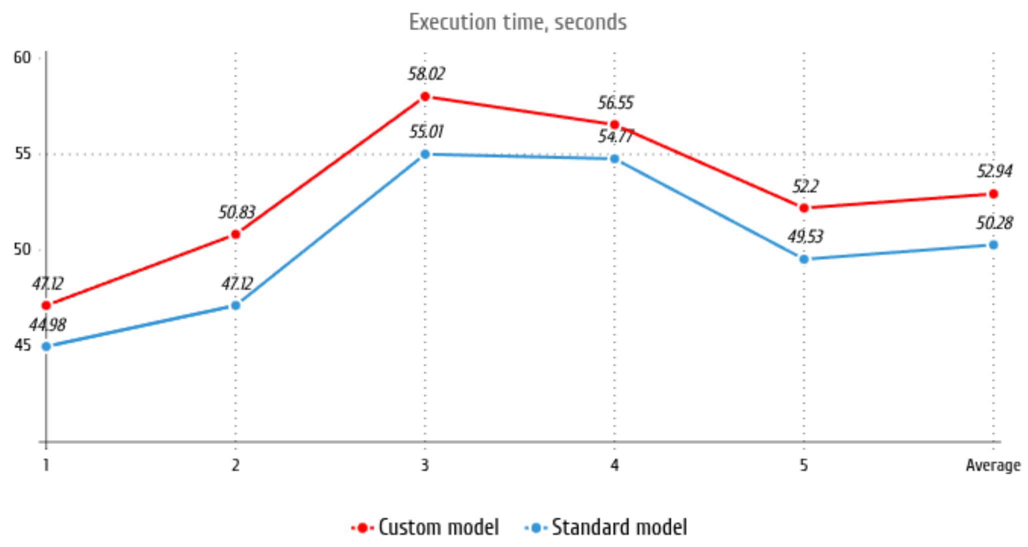

Fig. 4. Diagram illustrating the impact of a custom model on image generation time

The conducted experiments confirm that the combination of an asynchronous architecture, event-driven communication, and support for custom model training ensures not only the functional correctness of the system, but also its engineering suitability for scalability and practical deployment. The proposed approach enables the integration of artificial intelligence services into web applications without compromising system stability, while maintaining a balance between performance, output quality, and usability.

Conclusions and Prospects for Further Research. This study investigates an approach to integrating the Novita AI service into a web application with support for asynchronous request processing and custom model training. The proposed architecture provides scalability, stability, and controllability of the image generation processes under varying load conditions. Although Novita AI is used as a representative implementation case, the proposed architectural approach is applicable to other AI service platforms that provide API-based access and support asynchronous execution.

The experimental results demonstrate that resource-intensive operations, such as image generation and model training, are more efficiently handled using asynchronous processing, whereas lightweight operations can be performed synchronously without negatively affecting the user experience. Although the use of custom models slightly increases generation time, it significantly improves output quality and consistency, while reducing the need for manual parameter tuning.

The obtained results confirm the engineering feasibility of the proposed approach for practical application in web-based systems for automated visual content generation. Future research should focus on optimizing the model training process, supporting multimodal scenarios, and analyzing system behavior under peak load conditions.

References

1. Cao Y., Chen Z., Li S., Zhang Y. "A Survey on Generative Diffusion Models." *arXiv*, 2022, URL: https://doi.org/10.48550/arXiv.2209.02646.
2. Sordo, A., et al. A Review on Generative AI For Text-To-Image and Image-To-Image Generation and Implications To Scientific Images. arXiv:2502.21151, 2025. URL: https://doi.org/10.48550/arXiv.2502.21151.
3. Lishchyna N. M., Lishchyna V. O., Povstiana Yu. S. Approaches and algorithms for processing and image recognition of complex structure. Computer-Integrated Technologies: Education, Science, Production, № 38, 2020, pp. 5-9. DOI:10.36910/6775-2524-0560-2020-38-01.
4. Novita AI – Model Libraries & GPU Cloud, Deploy, Scale & Innovate. URL: https://novita.ai.
5. Novita AI – Serverless GPU Platform – Kodus. URL: https://docs.kodus.io/cookbook/en/novita.
6. Novita AI Text to Image API: Generate Stunning Visuals from Text with Stable Diffusion. URL: https://novita.ai/docs/api-reference/model-apis-text-to-image.
7. Gupta A., Chaturvedi Y. Cloud-Native ML: Architecting AI Solutions for Cloud-First Infrastructures. Nanotechnology Perceptions, vol. 20, no. 7, 2024, pp. 930–939. DOI: 10.62441/nano-ntp.v20i7.4004.
8. Saxena S., Teli M.N. Comparison and Analysis of Image-to-Image Generative Adversarial Networks: A Survey. arXiv. URL: https://doi.org/10.48550/arXiv.2112.12625.
9. Godinho A., Rosado J., Sá F., Cardoso F. Method for Evaluating the Performance of Web-Based APIs. In: Smart Objects and Technologies for Social Good, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (LNICST), vol. 556. 2024. DOI: 10.1007/978-3-031-52524-7_3.

Історія статті:
Отримано: 16.02.2026 Доопрацьовано: 23.02.2026 Прийнято до друку: 23.03.2026 Опубліковано: 29.03.2026