*International periodic scientific journal*

ONLINE

www.sworldjournal.com

D.A.Tsenov Academy of Economics - Svishtov (Bulgaria)

Indexed in
**INDEXCOPERNICUS**
**(ICV: 73)**
GOOGLESCHOLAR

# SWorld Journal

**Issue №39**
**Part 1**
**September 2026**

*Published by:*
*SWorld & D.A. Tsenov Academy of Economics, Svishtov,Bulgaria*

**ISSN 2663-5712**
**DOI: 10.30888/2663-5712**

**Editor:** Shibaev Alexander Grigoryevich, *Doctor of Technical Sciences, Professor, Academician*
**Scientific Secretary:** Kupriienko Sergiy, *PhD in Technical Sciences*

**Editorial board:** More than 400 doctors of science. Full list on page:
*https://www.sworldjournal.com/index.php/swj/about/editorialTeam*

**Expert-Peer Review Board of the journal:** Full list on page:
*https://www.sworldjournal.com/index.php/swj/expertteam*

**T**he International Scientific Periodical Journal "SWorldJournal" has gained considerable recognition among domestic and foreign researchers and scholars. Today, the journal publishes authors from from different countries.

**J**ournal Established in 2018. **P**eriodicity of publication: 6 times a year

**T**he journal activity is driven by the following objectives:

- Broadcasting young researchers and scholars outcomes to wide scientific audience
- Fostering knowledge exchange in scientific community
- Promotion of the unification in scientific approach
- Creation of basis for innovation and new scientific approaches as well as discoveries in unknown domains

**T**he journal purposefully acquaints the reader with the original research of authors in various fields of science, the best examples of scientific journalism.

**P**ublications of the journal are intended for a wide readership - all those who love science. The materials published in the journal reflect current problems and affect the interests of the entire public.

**E**ach article in the journal includes general information in English.

**T**he journal is registered in the INDEXCOPERNICUS, GoogleScholar.

**DOI: 10.30888/2663-5712.2026-39-01**

**Published by:**
**SWorld &**
**D.A. Tsenov Academy of Economics**
*Svishtov, Bulgaria*
e-mail: editor@sworldjournal.com

Copyright
© Authors, scientific texts 2026

https://www.sworldjournal.com/index.php/swj/article/view/swj39-01-006
DOI: 10.30888/2663-5712.2026-39-01-006

**УДК 004.78:004.415:519.876**

# ARCHITECTURAL MODEL OF AN INFORMATION SYSTEM FOR THE AUTOMATED PROCESSING OF CUSTOMER INQUIRIES

## АРХІТЕКТУРНА МОДЕЛЬ ІНФОРМАЦІЙНОЇ СИСТЕМИ АВТОМАТИЗОВАНОЇ ОБРОБКИ ПОТОКУ КЛІЄНТСЬКИХ ЗВЕРНЕНЬ

**Brynza N.O./ Бринза Н.О.**
*к.т.н., доц.*
*ORCID:https://orcid.org/0000-0002-0229-2874*
*Simon Kuznets Kharkiv National University of Economics /*
*Харківський національний економічний університет ім. С. Кузнеця,*
*Харків, просп. Науки, 9-А, 61165,*
*Kharkiv National University of Radio Electronics /*
*Харківський національний університет радіоелектроніки,*
*Харків, просп. Науки 14, 61023*
**Mazepa A. S. / Мазепа А. С.**
*Аспірант*
*ORCID: https://orcid.org/0009-0006-9574-3174*
*Kharkiv National University of Radio Electronics /*
*Харківський національний університет радіоелектроніки,*
*Харків, просп. Науки 14, 61023*
**Zadrykin S. O. / Задрикін С. О.**
*Аспірант*
*ORCID: https://orcid.org/0009-0003-0977-0925*
*Kharkiv National University of Radio Electronics /*
*Харківський національний університет радіоелектроніки,*
*Харків, просп. Науки 14, 61023*

***Анотація.*** *У статті розглянуто проблему автоматизації обробки потоку клієнтських звернень, що надходять до компаній через різнорідні канали комунікації. Метою дослідження є розроблення та обґрунтування архітектурної моделі інформаційної системи, орієнтованої на автоматизоване приймання, реєстрацію, класифікацію, пріоритезацію та маршрутизацію клієнтських звернень. Для досягнення поставленої мети визначено функціональні етапи життєвого циклу звернення, сформовано склад основних компонентів системи та обґрунтовано їх взаємодію на основі мікросервісного та подійно-орієнтованого підходів. Запропоновано використання API Gateway для уніфікації взаємодії із зовнішніми каналами, асинхронних черг для організації потокової обробки та окремих сервісів для керування зверненнями, їх класифікації, пріоритезації, маршрутизації та сповіщення користувачів. Для оцінювання ефективності запропонованого архітектурного рішення визначено систему показників, що охоплює час первинної реакції, час обробки, пропускну здатність, завантаження операторів та частку необроблених звернень. Обґрунтовано можливість інтеграції AI/NLP-компонентів для автоматичної класифікації звернень, визначення їх пріоритету та формування рекомендацій щодо відповіді. Запропонована архітектурна модель може бути використана як концептуальна основа проєктування інформаційних систем підтримки клієнтів із можливістю подальшого масштабування та розширення функціональності.*

***Ключові слова:*** *інформаційна система, клієнтські звернення, мікросервісна архітектура, потокова обробка, автоматична маршрутизація, api gateway, асинхронна обробка, теорія масового обслуговування, штучний інтелект.*

**Вступ**

Цифровізація бізнес-процесів та розширення каналів взаємодії з клієнтами зумовлюють постійне збільшення обсягів інформації, що надходить до компаній у вигляді запитів, повідомлень, скарг, консультацій та інших видів клієнтських звернень. Сучасна взаємодія з клієнтами здійснюється через вебформи, електронну пошту, месенджери, соціальні мережі, контакт-центри та інші цифрові канали. Внаслідок цього інформаційна система обслуговування клієнтів має функціонувати не лише як засіб реєстрації окремих звернень, а як середовище централізованого приймання та послідовної обробки динамічного потоку різнорідних інформаційних повідомлень.

Зростання кількості каналів комунікації ускладнює організацію роботи з клієнтськими зверненнями. За відсутності єдиного механізму їх реєстрації та маршрутизації інформація може накопичуватися у різних інформаційних системах, що ускладнює формування цілісної історії взаємодії з клієнтом. Використання ручної або частково автоматизованої обробки також потребує виконання оператором значної кількості рутинних операцій, пов'язаних із первинною реєстрацією, визначенням категорії звернення, встановленням його пріоритету та передачею відповідальному працівнику. Це створює передумови для збільшення часу первинної реакції, нерівномірного розподілу навантаження та виникнення помилок під час обробки інформації.

Окремою проблемою є динамічний характер надходження звернень. Інтенсивність вхідного потоку може змінюватися залежно від часу доби, маркетингових активностей, змін у роботі сервісів, виникнення технічних інцидентів або інших зовнішніх факторів. За таких умов статичний розподіл звернень між працівниками не завжди забезпечує раціональне використання доступних ресурсів. Тому інформаційна система повинна враховувати не лише зміст і характеристики конкретного звернення, а й поточний стан черг, доступність операторів, їх компетенції та встановлені вимоги до часу обслуговування.

Традиційні системи управління взаємодією з клієнтами, Service Desk та

Ticketing Systems забезпечують значну частину функцій, необхідних для реєстрації та контролю звернень. Водночас їх функціональні можливості та архітектурні рішення можуть відрізнятися залежно від призначення системи, способів інтеграції із зовнішніми каналами та вимог до масштабування. Для систем, орієнтованих на обробку динамічного потоку звернень, особливого значення набувають можливості асинхронної взаємодії, незалежного масштабування функціональних компонентів, автоматизованої маршрутизації та оперативного моніторингу стану обробки.

З архітектурної точки зору зазначені вимоги обґрунтовують доцільність застосування мікросервісного та подійно-орієнтованого підходів. Декомпозиція функціональності на окремі сервіси дає змогу розмежувати відповідальність за приймання, реєстрацію, класифікацію, пріоритезацію, маршрутизацію та супровід звернень. Використання асинхронних черг, своєю чергою, дає можливість розділити процес надходження та подальшої обробки повідомлень, що створює передумови для більш гнучкого керування навантаженням і масштабування окремих компонентів системи [1].

При цьому архітектурна побудова системи обробки клієнтських звернень повинна враховувати взаємозв'язок між програмними компонентами, інформаційними потоками та ресурсами, задіяними у процесі обслуговування. Тому доцільним є розгляд такої системи не лише як сукупності програмних модулів, а як інформаційної системи, функціонування якої може бути формалізоване з використанням показників теорії масового обслуговування та показників якості обслуговування. До таких показників належать час первинної реакції, середній час обробки, пропускна здатність, рівень завантаження операторів та частка звернень, обробка яких не була завершена у встановлений термін.

Наукова проблема полягає у формуванні архітектурного рішення, яке забезпечує узгоджену автоматизацію повного циклу роботи з клієнтським зверненням – від його надходження через різнорідні канали до класифікації, визначення пріоритету, маршрутизації, виконання та збереження історії

взаємодії – з урахуванням динаміки вхідного потоку та можливості масштабування окремих функціональних компонентів. Додатковим напрямом розвитку такої архітектури є інтеграція інтелектуальних компонентів для автоматизації операцій, пов'язаних з аналізом текстового змісту звернень, без критичної залежності основних функцій системи від доступності AI-сервісів.

Актуальним є розроблення архітектурної моделі інформаційної системи автоматизованої обробки потоку клієнтських звернень, яка поєднує омніканальне приймання інформації, мікросервісну декомпозицію, асинхронну взаємодію компонентів, автоматизовану маршрутизацію та засоби оцінювання ефективності обробки. Такий підхід створює концептуальну основу для подальшого проєктування та практичної реалізації систем підтримки клієнтів із можливістю адаптації до зміни інтенсивності вхідного потоку.

**Аналіз сучасних підходів**

Проблема автоматизації обробки звернень досліджувалася в контексті створення CRM-систем, платформ Service Desk / Help Desk та систем управління інцидентами. Сучасні CRM-системи орієнтовані на ведення історії взаємозв'язків та продажів, проте нерідко мають обмежені можливості щодо високошвидкісної потокової обробки неструктурованих даних у реальному часі. Системи Service Desk чудово формалізують процеси управління інцидентами та запитами на обслуговування, але є відносно монолітними, складними в налаштуванні та інтеграції з локальними месенджерами чи нестандартними вебканалами.

Системи управління зверненнями забезпечують базовий функціонал створення квитків та їх прив’язки до операторів, однак у класичній реалізації часто функціонують як розширені бази даних, не надаючи розвинених інструментів динамічного балансування потоків та інтелектуальної маршрутизації.

Аналіз сучасних архітектурних парадигм показує, що перехід від монолітних побудов до мікросервісної архітектури, яка взаємодіє через REST API та подійно-орієнтовані шини даних, дозволяє усунути ключові обмеження класичних систем. Мікросервісний підхід забезпечує незалежне масштабування

окремих компонентів, високу відмовостійкість та легкість інтеграції із зовнішніми базами даних і сервісами штучного інтелекту [2].

Обґрунтованим є синтез переваг омніканальної взаємодії, мікросервісної побудови програмного забезпечення та концепцій подійно-орієнтованого керування потоками для створення спеціалізованої архітектури інформаційної системи обробки звернень.

Метою статті є розроблення та науково-технічне обґрунтування багаторівневої мікросервісної архітектури інформаційної системи автоматизованої обробки потоку клієнтських звернень, орієнтованої на скорочення часу первинної реакції, балансування навантаження на операторів та адаптацію до коливань інтенсивності вхідного потоку.

**Функціональна модель процесу автоматизованої обробки клієнтського звернення**

Функціональна модель запропонованої інформаційної системи визначається послідовністю операцій, які забезпечують перетворення вхідного клієнтського повідомлення на структурований об'єкт обробки та його подальше передавання відповідальному виконавцю. На відміну від розгляду окремого звернення як ізольованої інформаційної одиниці, запропонований підхід передбачає організацію його обробки як складової динамічного потоку повідомлень, що надходять через різні канали комунікації (рис. 1).




**Рисунок 1 - Процес обробки звернення**

Такий порядок не передбачає обов'язкового виконання всіх операцій виключно послідовно. Частина процедур може виконуватися асинхронно або

паралельно, що особливо важливо для систем, які обробляють значну кількість одночасних звернень.

Першим етапом є отримання звернення через один із доступних каналів комунікації: вебформу, електронну пошту, месенджер, зовнішній API або інший інтеграційний сервіс. Оскільки різні канали можуть використовувати різні формати даних, первинною функцією системи є приведення отриманої інформації до уніфікованого внутрішнього представлення.

До такого представлення доцільно включати ідентифікатор звернення, ідентифікатор клієнта, дату та час надходження, джерело звернення, текст повідомлення, вкладені файли, категорію, поточний статус та інші атрибути, необхідні для подальшої обробки.

Уніфікація вхідних даних дає змогу відокремити особливості конкретного каналу комунікації від внутрішніх процесів системи. У результаті подальші функціональні компоненти працюють зі стандартизованим об'єктом звернення незалежно від способу його надходження.

На наступному етапі система визначає, чи існує відповідний профіль клієнта у внутрішньому сховищі даних. Ідентифікація може здійснюватися за обліковим записом, адресою електронної пошти, номером телефону, ідентифікатором зовнішньої системи або іншим доступним ідентифікатором.

Після ідентифікації формується або оновлюється запис про звернення. При цьому кожному зверненню присвоюється унікальний ідентифікатор, який використовується для відстеження його стану протягом усього життєвого циклу. Одночасне збереження інформації про клієнта та історії його попередніх звернень створює основу для подальшої персоналізованої обробки.

Зареєстроване звернення передається до функціонального компонента первинної обробки. Його завданням є визначення основних характеристик звернення, необхідних для подальшого прийняття рішення щодо способу обслуговування.

Однією з ключових операцій є класифікація звернення за тематикою або типом проблеми. Класифікація може здійснюватися за наперед визначеними

правилами, за допомогою алгоритмів машинного навчання або комбінованого підходу. Застосування методів автоматичної класифікації до текстових звернень дає змогу здійснювати їх тематичне групування та підвищувати ефективність подальшої обробки. Результатом є категорія звернення, яка використовується як один із параметрів його подальшої маршрутизації.

За необхідності на цьому етапі також можуть визначатися ознаки дублювання звернення, наявність ключових слів, пов'язаних із критичними ситуаціями, та інші характеристики, що впливають на порядок подальшої обробки.

Наступним етапом є встановлення пріоритету звернення. Пріоритет визначається відповідно до сукупності характеристик, серед яких можуть враховуватися категорія проблеми, критичність ситуації, характеристики клієнта, час очікування та встановлені правила обслуговування.

Формально пріоритет звернення $p_i$ можна представити як функцію його параметрів:

$$\mathrm{p}_i = f(c_i, s_i, t_i, r_i)$$

де $c_i$ – категорія звернення; $s_i$ – рівень критичності; $t_i$ – тривалість очікування; $r_i$ – додаткові правила або характеристики клієнта.

Таке представлення не задає конкретного алгоритму розрахунку пріоритету, а визначає загальний принцип, відповідно до якого порядок обробки формується на основі сукупності параметрів звернення.

Після визначення категорії та пріоритету звернення передається до механізму маршрутизації. Його призначенням є вибір відповідного підрозділу, черги або конкретного оператора, який має виконувати подальшу обробку.

Для прийняття рішення можуть використовуватися такі характеристики: категорія та тематика звернення; спеціалізація оператора; поточне навантаження; доступність оператора; пріоритет звернення; встановлені правила обслуговування; час очікування звернення у черзі.

Маршрутизація розглядається як функція, що пов'язує характеристики вхідного звернення зі станом доступних ресурсів системи. У спрощеному вигляді

результат маршрутизації можна представити як:

$$a_i = arg \max_{a \epsilon A} F(x_i, a)$$

де $x_i$ – сукупність характеристик *i*-го звернення, $A$ – множина доступних виконавців або черг, а $F(x_i, a)$ – функція відповідності звернення конкретному виконавцю.

Запропонований підхід дає змогу перейти від статичного розподілу звернень до динамічної маршрутизації, яка враховує поточний стан системи.

Після маршрутизації звернення переходить до відповідального виконавця. На цьому етапі здійснюється безпосереднє вирішення проблеми або надання необхідної інформації клієнту.

Система повинна забезпечувати можливість зміни статусу звернення та фіксації ключових подій його обробки. Це дає змогу сформувати послідовну історію взаємодії та контролювати дотримання встановлених термінів обслуговування.

Формування відповіді може здійснюватися оператором вручну або за підтримки автоматизованих засобів. Зокрема, AI/NLP-компоненти можуть використовуватися для формування рекомендацій щодо відповіді, пошуку аналогічних випадків або підготовки шаблонів, які надалі перевіряються та за необхідності коригуються оператором.

Після надання відповіді звернення переводиться у відповідний завершальний статус. При цьому система зберігає інформацію про основні події його життєвого циклу: час надходження, результати класифікації та маршрутизації, зміни статусу, час початку та завершення обробки, а також результати взаємодії з клієнтом.

Накопичена історія використовується не лише для повторного обслуговування клієнта, а й для формування аналітичних показників. На її основі можуть визначатися середній час первинної реакції, середній час обробки, кількість звернень за категоріями, навантаження операторів, частка повторних звернень та інші характеристики функціонування системи.

Принциповою особливістю запропонованої моделі є розгляд звернень як потоку інформаційних об'єктів, для якого інтенсивність надходження та доступність ресурсів можуть змінюватися у часі. Тому між етапами приймання та обробки доцільно передбачити механізм буферизації звернень, що дає змогу розділити швидкість їх надходження та швидкість обслуговування.

У такій моделі вхідний потік характеризується інтенсивністю $\lambda$, тоді як продуктивність одного оператора або іншого ресурсу обслуговування може бути представлена інтенсивністю $\mu$ [3]. За наявності $n$ паралельних виконавців коефіцієнт їх сукупного завантаження може бути оцінений співвідношенням:

$$\rho = \frac{\lambda}{n\mu}.$$

Значення цього показника може використовуватися для оцінювання відповідності доступної пропускної здатності поточній інтенсивності надходження звернень. Водночас для практичної оцінки системи необхідно враховувати нерівномірність потоку, різну складність звернень, пріоритети та можливість повторного надходження повідомлень.

Функціональна модель визначає не лише послідовність операцій над окремим зверненням, а й принцип організації їх колективної обробки у динамічному потоці. Запропонована модель є основою для формування архітектури системи, оскільки кожному ключовому функціональному етапу може відповідати окремий сервіс або група взаємопов'язаних компонентів.

**Архітектура системи.**

Для забезпечення високої доступності [4], незалежного масштабування та гнучкості інтеграцій розроблено багаторівневу мікросервісну архітектуру системи автоматизованої обробки потоку клієнтських звернень, схему якої зображено на рис. 2.

Основні функціональні блоки та компоненти запропонованої архітектури подано у таблиці 1.

Взаємодія між мікросервісами реалізується у гібридному режимі: синхронний доступ за протоколом REST API / gRPC використовується для

оперативних CRUD-операцій, тоді як передача звернення між етапами обробки виконується асинхронно через брокер повідомлень [5], що дає змогу буферизувати повідомлення під час короткочасних сплесків трафіку та розділити швидкість надходження і обробки звернень.

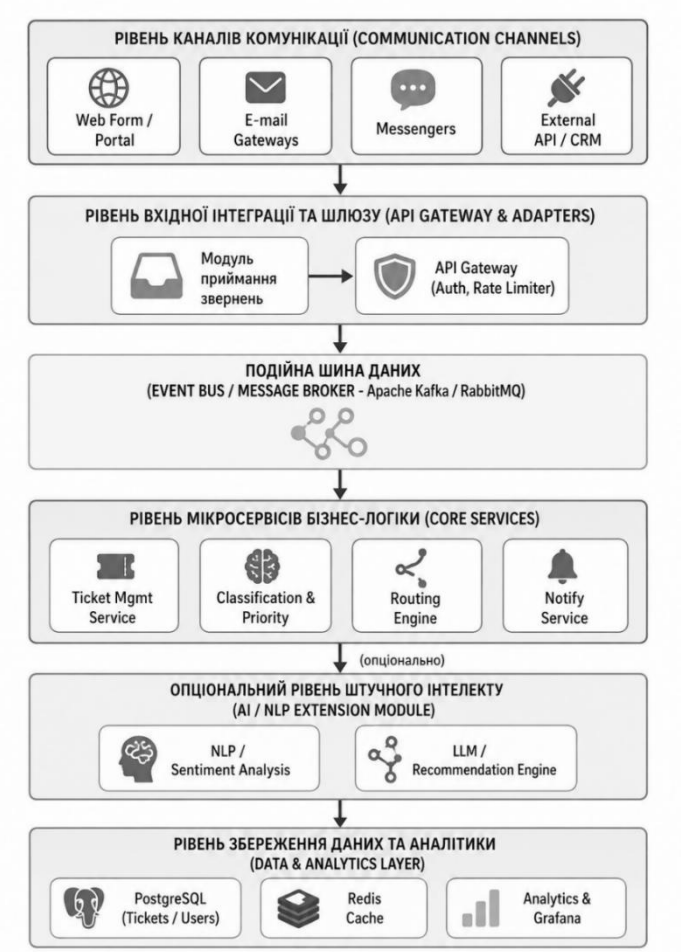



**Рисунок 2 - Концептуальна мікросервісна архітектура інформаційної системи автоматизованої обробки потоку клієнтських звернень**

**Таблиця 1 - Функціональні компоненти системи та їх призначення**

| Назва компонента | Тип / Стек | Функціональне призначення та роль у системі |
|---|---|---|
| Модуль приймання звернень | Адаптери каналів | Забезпечує уніфікований прийом повідомлень з e-mail, месенджерів та вебформ, вилучення метаданих та конвертацію в єдиний JSON-формат. |
| API Gateway | Reverse Proxy / API Management | Слугує єдиною точкою входу, здійснює автентифікацію, маршрутизацію HTTP-запитів, обмеження частоти запитів та балансування навантаження. |

| | | |
|---|---|---|
| Сервіс керування зверненнями | Core Microservice (REST/gRPC) | Забезпечує повний життєвий цикл звернення |
| Сервіс класифікації та пріоритезації | Business Logic Service | Аналізує параметри звернення, застосовує бізнес-правила або AI-моделі для призначення категорії, тегів та рівня SLA. |
| Сервіс маршрутизації | Dispatcher Microservice | Реалізує алгоритми Skill-Based та Load-Balanced Routing. Обирає оптимального оператора з урахуванням поточного стану черг та компетенцій. |
| Сервіс сповіщень | Notification Service | Формує та надсилає тригерні повідомлення клієнтам та операторам про зміну стану звернення. |
| База даних (БД) | PostgreSQL + Redis | Реляційне сховище для структурованих даних та In-Memory кєш для оперативних черг і сесій. |
| Модуль аналітики та моніторингу | OLAP / TimescaleDB + Grafana | Агрегує часові метрики, будує аналітичні звіти щодо FRT, AHT, CSAT, навантаження операторів та здійснює моніторинг Health-Check. |

**Організація обробки потоку звернень**

Особливістю функціонування розробленої системи є робота з динамічним потоком вхідних запитів. Інтенсивність потоку $\lambda(t)$ може суттєво коливатися протягом доби або внаслідок виникнення інцидентів. Для забезпечення стійкості застосовується концепція систем масового обслуговування [6].

Вхідний потік звернень поступає у FIFO/Priority чергу. Система здійснює аналіз та обробку повторних звернень і дублікатів за допомогою хешування тексту та перевірки часового. Якщо час перебування звернення у черзі перевищує поріг SLA, Сервіс керування автоматично запускає процедуру ескалації.

Для математичного формалізму та оцінювання ефективності системи запропоновано комплекс показників (табл. 2).

Балансування навантаження реалізується за рахунок горизонтального автомасштабування мікросервісів при збільшенні довжини черги в Message Broker.

**Таблиця 2 - Показники оцінювання ефективності системи обробки звернень**

| Показник | Позначення / Порядок | Формалізація та системний зміст |
|---|---|---|
| Середній час первинної реакції | $T_{FRT}$ (хв / сек) | $T_{FRT} = \frac{1}{N}\sum(t_{first_resp} - t_{arrival})$. Визначає оперативність системи на етапі від прийому до першого контактного повідомлення. |
| Середній час обробки | $T_{AHT}$ (хв) | $T_{AHT} = \frac{1}{N}\sum(t_{end} - t_{start})$. Оцінює тривалість безпосередньої роботи оператора або автоматичного сервісу над зверненням. |
| Коефіцієнт завантаження операторів | $\rho(0\ldots1)$ | $\rho = \frac{\lambda}{n\cdot\mu}$, де $\lambda$ – інтенсивність потоку, n – кількість операторів, $\mu$ – інтенсивність обслуговування одним оператором. |
| Пропускна здатність системи | *S* (зверн/год) | Максимальна кількість звернень, яку система здана обробити за одиницю часу без порушення граничних термінів SLA. |
| Частка необроблених звернень | $P_{loss}$ (%) | $P_{loss} = \frac{N_{lost}}{N_{total}} \cdot 100\%$. Відношення втрачених або прострочених звернень до загальної кількості вхідного потоку. |

**Використання AI-компонентів у системі обробки клієнтських звернень**

Штучний інтелект у запропонованій інформаційній системі розглядається як додатковий функціональний компонент, призначений для автоматизації операцій, пов'язаних з аналізом змісту клієнтських звернень та підтримкою прийняття рішень під час їх подальшої обробки. При цьому AI-компоненти не повинні бути критично необхідними для виконання базових функцій системи. У разі тимчасової недоступності відповідного сервісу основний процес приймання, реєстрації, збереження та маршрутизації звернень має продовжувати функціонувати на основі стандартних правил.

У межах запропонованої архітектури доцільно виділити три основні напрями використання AI: автоматичну класифікацію звернень, інтелектуальне визначення пріоритету та формування рекомендацій для оператора.

Після реєстрації звернення його текстовий вміст може передаватися до NLP-компонента для визначення тематичної категорії. Результатом роботи моделі є клас звернення, який надалі використовується механізмом маршрутизації для вибору відповідної черги або групи операторів.

У спрощеному вигляді задача класифікації може бути представлена як:

$$c_i = f_{\text{NLP}}(x_i),$$

де $x_i$ – текстовий вміст *i*-го звернення, $f_{\text{NLP}}$ – модель автоматичної класифікації, $c_i$ – визначена категорія звернення.

Для реалізації цього функціоналу можуть застосовуватися як традиційні методи обробки тексту та машинного навчання [7], так і трансформерні мовні моделі. Вибір конкретної моделі залежить від обсягу навчальної вибірки, кількості категорій, вимог до швидкості обробки та доступних обчислювальних ресурсів.

Важливою особливістю архітектури є те, що результат класифікації передається не безпосередньо оператору, а до наступного етапу обробки – механізму визначення пріоритету та маршрутизації. Таким чином, AI-компонент інтегрується у загальний процес обробки звернення, не порушуючи його функціональної послідовності.

Другим напрямом використання AI є визначення пріоритету звернення. На відміну від статичного правила, за яким усі звернення певної категорії отримують однаковий рівень пріоритету, інтелектуальна модель може враховувати сукупність характеристик повідомлення та контексту його надходження.

До вхідних параметрів можуть належати категорія звернення, текст повідомлення, ознаки критичності проблеми, тривалість очікування, попередня історія взаємодії та інші параметри, передбачені бізнес-правилами системи.

Результатом є оцінка пріоритету:

$$p_i = f_{\text{AI}}(x_i, z_i),$$

де $x_i$ – текстові характеристики звернення, $z_i$ – додаткові структуровані параметри, а $p_i$ – визначений рівень пріоритету.

Практично доцільним є використання комбінованої схеми, у якій AI-модель формує оцінку, а остаточне рішення коригується відповідно до встановлених бізнес-правил. Такий підхід дає змогу зменшити залежність системи від виключно автоматичного рішення та врахувати критичні умови, які мають бути задані детермінованими правилами.

Третім напрямом є використання генеративних мовних моделей для підтримки оператора. На цьому етапі AI не замінює працівника, а виконує функцію інтелектуального помічника.

На основі тексту поточного звернення, його категорії та доступної історії взаємодії система може сформувати рекомендацію щодо відповіді, запропонувати відповідний шаблон або виділити інформацію, необхідну для вирішення проблеми.

Архітектурно цей компонент доцільно реалізувати як окремий AI-сервіс, взаємодія з яким здійснюється через API. Такий підхід забезпечує слабку зв'язаність AI-компонента з основними сервісами системи та дає можливість змінювати або модернізувати використовувану модель без перебудови базової архітектури (рис. 3).

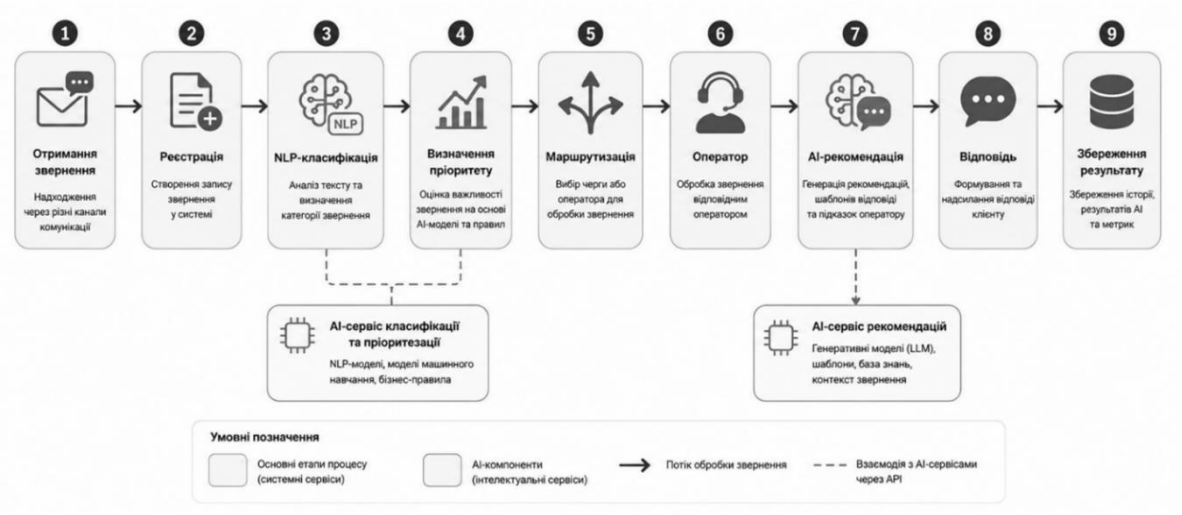



**Рисунок 3 - Загальна схема взаємодії AI-компонентів**

При цьому AI-сервіси взаємодіють із системою через визначені програмні інтерфейси та не повинні безпосередньо змінювати критичні дані без контролю

основних сервісів. Результати роботи моделей зберігаються разом з інформацією про відповідне звернення, що створює можливість подальшого аналізу якості класифікації, пріоритезації та рекомендацій.

Окремою вимогою є можливість обробки ситуацій, у яких AI-модель не може сформувати достовірний результат. У такому випадку система повинна передбачати механізм переходу до резервного сценарію: використання стандартних правил класифікації, передавання звернення до загальної черги або ручного визначення категорії оператором.

AI-компоненти доцільно розглядати як розширення функціональності системи, а не як її єдиний механізм прийняття рішень. Їх інтеграція у вигляді окремих слабкозв'язаних сервісів забезпечує можливість поступового впровадження інтелектуальних функцій, заміни моделей та адаптації системи до зміни характеру клієнтських звернень без принципової зміни базової архітектури.

**Висновки**

У результаті проведеного дослідження сформовано архітектурну модель інформаційної системи автоматизованої обробки потоку клієнтських звернень, орієнтовану на централізоване приймання, структуризацію, класифікацію, пріоритезацію та маршрутизацію звернень, що надходять через різнорідні канали комунікації. Запропонована модель розглядає звернення не як окремі інформаційні об'єкти, а як динамічний потік, інтенсивність якого може змінюватися у часі, що визначає необхідність застосування механізмів буферизації, асинхронної взаємодії та адаптивного розподілу навантаження.

Функціональну основу системи сформовано у вигляді послідовності взаємопов'язаних етапів: надходження звернення, ідентифікація клієнта, реєстрація, первинна обробка та класифікація, визначення пріоритету, маршрутизація, виконання, формування відповіді та збереження історії взаємодії. Така декомпозиція забезпечує відповідність між функціональними процесами та окремими архітектурними компонентами системи й створює основу для їх незалежного розвитку та масштабування.

Обґрунтовано доцільність використання мікросервісного та подійно-орієнтованого підходів для побудови системи. Мікросервісна декомпозиція дає змогу розмежувати відповідальність між компонентами, а асинхронна взаємодія через механізм обміну повідомленнями – розділити процес надходження та обробки звернень. Це створює передумови для буферизації повідомлень під час нерівномірного навантаження та незалежного масштабування окремих функціональних компонентів.

Для оцінювання ефективності функціонування системи сформовано набір показників, що охоплює середній час первинної реакції, середній час обробки, коефіцієнт завантаження ресурсів, пропускну здатність та частку необроблених або втрачених звернень. Застосування показників теорії масового обслуговування дає змогу пов'язати інтенсивність надходження звернень із доступною продуктивністю ресурсів обслуговування та оцінювати відповідність архітектури поточному навантаженню.

Окремим функціональним шаром системи визначено AI-компоненти, призначені для автоматичної класифікації звернень, інтелектуального визначення їх пріоритету та підтримки оператора під час формування відповіді. Запропоновано інтегрувати такі компоненти у вигляді слабкозв'язаних сервісів, що дає змогу замінювати або модернізувати використовувані моделі без принципової перебудови основних компонентів системи. При цьому базові операції приймання, реєстрації та збереження звернень не повинні залежати від доступності AI-сервісів, що підвищує стійкість архітектури до відмов окремих інтелектуальних компонентів.

Запропонована архітектурна модель поєднує функціональну декомпозицію процесу обробки клієнтських звернень, мікросервісний підхід, подійно-орієнтовану взаємодію, механізми динамічної маршрутизації та AI-підтримку прийняття рішень. Практичне значення отриманих результатів полягає у можливості використання запропонованої моделі як концептуальної основи для проєктування інформаційних систем підтримки клієнтів, здатних адаптуватися до зміни інтенсивності вхідного потоку та забезпечувати централізований

контроль життєвого циклу звернення.

Перспективним напрямом подальших досліджень є практична реалізація запропонованої архітектури та її експериментальне оцінювання на тестових і реальних потоках звернень. Особливу увагу доцільно приділити порівнянню різних стратегій маршрутизації, оцінюванню впливу AI-класифікації на час обробки та точність призначення звернень, а також визначенню параметрів масштабування системи залежно від інтенсивності вхідного потоку.

**Список використаних джерел:**

1. Minukhin S., Brynza N., Sitnikov D. (2021) Analyzing Performance of Apache Spark MLlib with Multinode Clusters on Azure HDInsight: Spark-Perf Case Study. In: Babichev S., Lytvynenko V., Wójcik W., Vyshemyrskaya S. (eds) Lecture Notes in Computational Intelligence and Decision Making. ISDMCI 2020. Advances in Intelligent Systems and Computing, vol 1246. Springer, Cham. https://doi.org/10.1007/978-3-030-54215-3_8.

2. Francesco, Paolo & Lago, Patricia & Malavolta, Ivano. (2019). Architecting with Microservices: a Systematic Mapping Study. Journal of Systems and Software. DOI: 150. 10.1016/j.jss.2019.01.001.

3. Lakatos, László & Szeidl, László & Telek, Miklós. (2013). Introduction to Queueing Systems with Telecommunication Applications. DOI: 10.1007/978-1-4614-5317-8.

4. Waseem, Muhammad & Liang, Peng & Shahin, Mojtaba. (2020). A Systematic Mapping Study on Microservices Architecture in DevOps. Journal of Systems and Software. DOI: 170. 110798. 10.1016/j.jss.2020.110798.

5. Laigner, Rodrigo & Almeida, Ana & Assunção, Wesley & Zhou, Yongluan. (2024). An Empirical Study on Challenges of Event Management in Microservice Architectures. DOI: 10.48550/arXiv.2408.00440.

6. Chakravarthy, Srinivas. (2022). Analysis of a Queueing Model with MAP Arrivals and Heterogeneous Phase-Type Group Services. Mathematics. DOI: 10.3575.10.3390/math10193575.

7. Bonechi, S., Palma, G., Caronna, M., Ugolini, M., Massaro, A., Rizzo, A. (2024). Enhancing Customer Support in Banking: Leveraging AI for Efficient Ticket Classification. PROCEDIA COMPUTER SCIENCE, 246(C), 128-137 DOI: 10.1016/j.procs.2024.09.235.

***Abstract.*** *This article examines the problem of automating the processing of customer inquiries received by companies through various communication channels. The aim of the study is to develop and justify an architectural model for an information system focused on the automated receipt, registration, classification, prioritization, and routing of customer inquiries. To achieve this goal, the functional stages of the inquiry lifecycle were identified, the set of key system components was defined, and their interaction was justified based on microservice and event-driven approaches. The use of an API Gateway is proposed to unify interaction with external channels, asynchronous queues to organize stream processing, and separate services to manage requests, classify them, prioritize them, route them, and notify users. To evaluate the effectiveness of the proposed architectural solution, a set of metrics has been defined, covering initial response time, processing time, throughput, agent workload, and the proportion of unprocessed requests. The possibility of integrating AI/NLP components for the automatic classification of requests, determining their priority, and generating response recommendations has been justified. The proposed architectural model can be used as a conceptual basis for designing customer support information systems with the potential for future scaling and functionality expansion.*

***Key words:*** *information system, customer requests, microservice architecture, stream processing, automatic routing, api gateway, asynchronous processing, queuing theory, artificial intelligence*

Стаття надіслана: 18.08.2026

© Бринза Н.О.