UDC 004.415.2:005.8:355.58

M.V. Miletskiy
A.T. Terenchuk

ARCHITECTURE OF VOLUNTEER AND RESOURCE COORDINATION PLATFORMS DURING EMERGENCY SITUATIONS

Vinnytsia National Technical University

Анотація

У тезах розглядається архітектура цифрових платформ для координації волонтерів та ресурсів під час надзвичайних ситуацій. Проаналізовано ключові архітектурні підходи, зокрема мікросервісна та подієво-орієнтована архітектура, що забезпечують масштабованість, відмовостійкість і гнучкість платформ. Описано основні модулі системи: управління волонтерами, розподіл ресурсів, геолокаційне відстеження та інтеграція з зовнішніми службами. Обговорюється вибір технологічного стеку та механізми забезпечення безпеки даних.

Ключові слова: координація волонтерів, надзвичайні ситуації, мікросервісна архітектура, розподіл ресурсів, платформа гуманітарної допомоги, подієво-орієнтована архітектура, геолокація.

Abstract

The paper examines the architecture of digital platforms for coordinating volunteers and resources during emergency situations. Key architectural approaches, including microservice and event-driven architecture, are analyzed to ensure platform scalability, fault tolerance, and flexibility. The main system modules are described: volunteer management, resource allocation, geolocation tracking, and integration with external services. The choice of technology stack and data security mechanisms are discussed.

Keywords: volunteer coordination, emergency situations, microservice architecture, resource allocation, humanitarian aid platform, event-driven architecture, geolocation.

Introduction

The increasing frequency and scale of natural disasters, armed conflicts, and other emergencies demand effective digital solutions for rapid response coordination. Traditional paper-based or phone-based approaches are insufficient for managing large numbers of volunteers and material resources in real time. In Ukraine and globally, the experience of recent crises has demonstrated that well-designed digital platforms can drastically reduce response time and improve the efficiency of humanitarian operations [1]. This paper addresses the architectural principles underlying platforms designed to coordinate volunteers and resources during emergencies.

Architectural Requirements and Design Principles

Platforms for emergency coordination must satisfy stringent non-functional requirements: high availability (99.9% uptime or better), horizontal scalability to handle sudden spikes in user load, and sub-second response latency for critical operations. Security and privacy of personal data, offline-first capabilities for unstable network conditions, and multi-tenancy support for different organizations are equally important [2-3]. From a design perspective, the separation of concerns principle suggests decomposing the system into clearly bounded contexts: user identity and authorization, volunteer lifecycle management, resource cataloguing and allocation, task dispatching, notifications, and geospatial services. Each context can evolve independently, reducing the risk of cascading failures.

Microservice and Event-Driven Architecture

A microservice architecture (MSA) is well suited for emergency coordination platforms because it allows individual services to be deployed, scaled, and updated independently. Each microservice owns its data store, communicating with peers via well-defined APIs or an asynchronous message broker such as Apache Kafka or RabbitMQ. This loose coupling ensures that a failure in one service, for example the notification service, does not prevent the resource allocation service from functioning [4]. Event-driven architecture (EDA)

complements MSA by enabling real-time data propagation. When a new task is created or a resource becomes available, an event is published to a topic. Interested consumers – mapping services, volunteer apps, analytics dashboards – react to this event without tight coupling. This pattern is particularly valuable during emergencies where the system state changes rapidly and all stakeholders must receive consistent, up-to-date information [5].

Core Platform Modules

The proposed architecture comprises the following principal modules: (1) Volunteer Management – handles registration, skill profiling, availability scheduling, and assignment tracking; (2) Resource Allocation – maintains an inventory of material assets (food, medicine, equipment) and supports automated matching algorithms; (3) Geolocation and Mapping – provides real-time GPS tracking of volunteers and vehicles via OpenStreetMap or similar services; (4) Task Dispatcher – applies priority-based scheduling to match volunteers with tasks by proximity, skill, and urgency; (5) Notification Gateway – delivers push, SMS, and email alerts via a unified API; (6) Analytics and Reporting – aggregates operational data for decision-makers and post-incident review [3; 6].

Technology Stack and Security

For the backend, Node.js or Go offer high throughput for I/O-bound coordination tasks, while Python suits analytics components. PostgreSQL with the PostGIS extension handles both relational and spatial queries. React and React Native enable code sharing across web and mobile surfaces. Container orchestration via Kubernetes provides automated scaling and self-healing, and a CI/CD pipeline with automated testing maintains reliability during continuous updates [2; 7]. Regarding security, OAuth 2.0 and OpenID Connect provide standards-compliant authentication, while role-based access control (RBAC) limits data exposure according to organizational hierarchy. All inter-service communication must be encrypted via TLS 1.3. Given the potential for targeted cyberattacks during crises, rate limiting, intrusion detection, and regular penetration testing are essential [1; 7].

Conclusions

The analysis shows that combining microservice and event-driven approaches, geospatial capabilities, and a modular design provides the best foundation for volunteer and resource coordination platforms in emergencies. The proposed architecture ensures high availability, scalability, and security while remaining flexible enough to integrate with existing governmental and NGO systems. Future work will focus on incorporating artificial intelligence for predictive resource allocation and automated task prioritization, further enhancing platform effectiveness in crisis response.

REFERENCES

1. Currion P. Information and Communication Technologies for Humanitarian Organizations. ALNAP. 2021. URL: https://www.alnap.org/ict-humanitarian (Accessed: May 25, 2026).
2. Newman S. Building Microservices: Designing Fine-Grained Systems. 2nd ed. Sebastopol: O'Reilly Media, 2021. 616 p.
3. Altay N., Labonte M. Challenges in Humanitarian Information Management and Exchange: Evidence from Haiti. Disasters. 2014. Vol. 38, No. S1. P. S50–S72. DOI: 10.1111/disa.12052.
4. Richardson C. Microservices Patterns: With Examples in Java. Shelter Island: Manning Publications, 2018. 520 p.
5. Stopford B. Designing Event-Driven Systems. Sebastopol: O'Reilly Media, 2018. 206 p.
6. Meier P. Digital Humanitarians: How Big Data Is Changing the Face of Humanitarian Response. Boca Raton: CRC Press, 2015. 252 p.
7. Burns B., Beda J., Hightower K. Kubernetes: Up and Running. 3rd ed. Sebastopol: O'Reilly Media, 2022. 326 p.

Мілецький Максим Васильович – студент групи 1ПІ-22б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e–mail: restar2801@gmail.com.

Теренчук Анатолій Тимофійович – кандидат технічних наук, доцент, старший викладач кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e–mail: anateren59@gmail.com.

Miletskiy Maksym V. – student of group 1PI-22b, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: restar2801@gmail.com.

Terenchuk Anatolii T. – Candidate of Technical Sciences, Associate Professor, Senior Lecturer of the Software Engineering Department, Vinnytsia National Technical University, Vinnytsia, e–mail: anateren59@gmail.com.