
AN INTEGRATED MOBILE FRAMEWORK FOR REDUCING EMERGENCY RESPONSE DELAYS USING REAL-TIME COORDINATION

Pillai Abin^{1*} and Kanojia Mahendra²¹Department of Computer Science, Sheth. L.U.J. and Sir M.V. College, India, pillaiabincs232414@gmail.com²Department of Computer Science, Sheth. L.U.J. and Sir M.V. College, India, kgkmahendra@gmail.com

*Corresponding author: Pillai Abin, Department of Computer Science, Sheth. L.U.J. and Sir M.V. College, India, pillaiabincs232414@gmail.com

ABSTRACT

Emergency medical services frequently encounter delays during the prehospital phase due to limited communication, human mediated prioritization, and sequential coordination among patients, dispatch centers, ambulances, and hospitals. These limitations are characteristic of call based emergency response workflows, which provides limited support for real-time information sharing, parallel decision making, and adaptive coordination under dynamic conditions. To address these structural challenges, this paper proposes an agent based mobile framework for reducing emergency response delays through real-time coordination. The proposed framework adopts a multi-agent architecture in which key stakeholders in the emergency response process are represented as autonomous yet coordinated software agents, including patient, ambulance/EMS, hospital, and orchestration agents. These agents interact through an event driven, publish–subscribe communication model supported by a cloud based coordination layer, enabling continuous state updates, parallel resource allocation, and synchronized decision flows. A layer of AI assisted decision support is integrated to assist with incident prioritization, decision support logic, ambulance assignment, and hospital selection. The primary contribution is a system level architectural design that formalizes coordination among emergency response components. This work presents a systematic design of agent roles, interaction mechanisms, operational workflows, and coordination logic for emergency response systems. The framework demonstrates how early incident assessment, parallelized ambulance hospital coordination, and real-time status synchronization may reduce non-clinical delays arising from manual communication and sequential decision making. By providing an integrated architectural blueprint for mobile, real-time emergency coordination, this work aims to inform future simulation studies, pilot implementations, and experimental evaluations seeking to improve the efficiency, scalability, and integration of prehospital emergency care systems.

Keywords: *emergency healthcare, multi-agent systems, real-time coordination, publish–subscribe communication, mobile health framework, cloud orchestration, AI-assisted decision support*

1. INTRODUCTION

Timely and well-coordinated prehospital emergency response is a critical determinant of survival and functional outcomes in time sensitive conditions such as trauma, cardiac arrest, and stroke, where delays of only a few minutes can substantially worsen prognosis (Kim & Oh, 2023; Zhang et al., 2020). Contemporary emergency medical services (EMS) increasingly emphasize integrated, patient centered care pathways that span the scene, transport, and receiving facility, highlighting the importance of early emergency assessment, rapid deployment of appropriate resources, and seamless information flow across organizational boundaries (Böbel et al., 2025; Cimino & Braun, 2023). Nevertheless, many systems still rely on manual, telephone based call handling, paper or verbally transferred emergency assessment information, and fragmented data systems, which together reduce situational awareness and timely decision making during the prehospital phase (Friesen et al., 2024; Garrido-Mancilla & Reynaldos-Grandón, 2023).

Despite decades of incremental improvements, traditional emergency response workflows remain constrained by telephony centric coordination, stage by stage transfers, and heterogeneous documentation practices that limit real time visibility of patient status, resource availability, and system load (Janerka et al., 2023; Rådestad et al., 2024). Human dependent decision making and dispatch decisions, often made under severe time pressure with incomplete information, can contribute to misprioritization and suboptimal destination choice, particularly when prehospital and hospital systems are weakly coupled (Kim & Oh, 2023; Widiyawati et al., 2025). Information exchange between patients, dispatch centers, EMS crews, and hospitals is frequently mediated through serial voice communication, which is vulnerable to loss, distortion, and delays and rarely supports structured feedback loops or automated decision support (Zhang et al., 2020; Smith et al., 2025). These structural limitations underscore the need for architectures that treat emergency response as a distributed, information intensive process requiring continuous, real-time coordination rather than a linear sequence of loosely connected steps.

Concurrently, the rapid diffusion of mobile technologies, the Internet of Medical Things (IoMT), telehealth platforms, and intelligent analytics has opened new possibilities for dynamic emergency assessment, continuous monitoring, and context aware coordination in prehospital care (Jasim et al., 2024; Janerka et al., 2023; Elfahim et al., 2025). Mobile phone based community response systems, geolocation enabled first responder applications, and prehospital telemedicine services illustrate how real-time data sharing and remote supervision can extend the reach of formal EMS and support earlier, situation aware interventions, including in resource constrained and rural settings (Friesen et al., 2024; Hahnenkamp et al., 2025; Smith et al., 2025). At the same time, studies of digital and telehealth solutions consistently report technical, usability, and organizational challenges, including interoperability issues, variable connectivity, and difficulties in integrating new tools into existing workflows, indicating that partial adoption of technology without coherent system design may fail to deliver sustained improvements (Zhang et al., 2020; Garrido-Mancilla & Reynaldos-Grandón, 2023).

Within this evolving landscape, multiagent systems and related paradigms for distributed intelligent coordination offer a promising foundation for rethinking prehospital emergency management. Prior work has shown that autonomous yet cooperating agents can represent key actors and services in prehospital processes, supporting complex tasks such as ambulance navigation, dynamic resource assignment, and real-time quality monitoring (Safdari et al., 2017). Event driven communication patterns and centralized orchestration mechanisms have also been proposed more broadly in healthcare to integrate heterogeneous data streams and align decisions across organizational units, particularly when rapid responses are required (Munagandla et al., 2024; Li et al., 2025). However, existing technological initiatives in prehospital and tele-emergency care typically focus on specific components such as teleconsultation, initial assessment, or data integration, rather than providing an integrated, agent based coordination framework that simultaneously connects patients, EMS, and hospitals through real-time, event driven workflows (Nanda et al., 2020; Damaševičius et al., 2023; Mustafa et al., 2025).

This paper addresses this gap by proposing an integrated mobile framework for reducing emergency response delays using real-time coordination, in an agent based architecture. The framework reconceptualizes prehospital emergency response as a network of interacting software agents representing key stakeholders and services, coordinated through event-driven communication and centralized orchestration to support continuous information sharing, dynamic initial assessment, and adaptive resource allocation across patients, EMS units, and receiving hospitals (Han & Choi, 2024; Reis et al., 2025; Zhang et al., 2025). The contribution of this work is architectural and methodological rather than empirical: the paper specifies the roles, interactions, and coordination mechanisms of the proposed multiagent system, articulates how mobile and real-time communication capabilities can be systematically integrated into prehospital workflows, and discusses design considerations and potential advantages relative to traditional telephony centric models. The framework is presented as a blueprint intended to guide future implementation, simulation based assessment, and pilot studies aimed at evaluating feasibility, safety, and effectiveness before consideration for real world deployment.

Although progress has been made in computer aided dispatch (CAD), mobile health (mHealth) applications, and ambulance routing optimization systems, these approaches typically improve isolated components of the emergency response chain rather than providing an integrated coordination architecture. Traditional CAD and dispatch technologies enhance call logging, incident tracking, and communication support but remain largely telephony-oriented and sequential in their operational logic (Friesen et al., 2024; Magnusson et al., 2024). mHealth applications facilitate user-side reporting, location sharing, and symptom documentation, yet they often operate as standalone tools without deep integration into live ambulance fleet management or hospital capacity systems (Roncero et al., 2020; Alharbi et al., 2020; Gabella et al., 2024). Similarly, ambulance tracking and routing optimization systems focus on route efficiency and resource positioning but generally do not incorporate structured patient input and real-time hospital readiness into a unified coordination process (Thajjiam, 2023; D et al., 2024; Teng et al., 2024). In contrast, the proposed framework introduces a unified multi-agent architecture that simultaneously integrates patient reporting, ambulance allocation, routing optimization, and hospital selection within a single event-driven ecosystem. Unlike conventional systems where ambulance dispatch and hospital coordination occur sequentially, the proposed architecture enables parallelized decision-making in which ambulance selection and hospital matching are executed based on real-time operational and clinical data. The primary novelty of this work lies in its system-level integration design. By modeling Patient, Ambulance/EMS, Hospital, and Orchestrator roles as interacting software agents communicating through a publish-subscribe mechanism, consistent with recent multi-agent coordination approaches in healthcare systems (Li et al., 2025; Han & Choi, 2024; Borghoff et al., 2025), the framework restructures emergency response from a linear communication chain into a synchronized, multi-stakeholder

coordination network. This architectural formalization of end-to-end real-time coordination distinguishes the proposed framework from existing CAD, mHealth, and routing solutions.

2. LITERATURE REVIEW

Emergency response coordination has traditionally relied on computer-aided dispatch (CAD) platforms designed to support call handling, incident logging, and ambulance allocation. Early CAD systems improved manual dispatch operations by digitizing call records and enabling basic unit tracking (Patel & Wang, 2016; Sousa et al., 2019). While these systems enhanced documentation and dispatcher oversight, their workflow logic remained largely telephone-oriented and sequential in nature (Chen et al., 2018; Aringhieri et al., 2017). Communication between dispatch centers, EMS crews, and hospitals frequently occurred via radio or voice calls, limiting structured data exchange and automated coordination (Friesen et al., 2024; Magnusson et al., 2024). Although contemporary CAD platforms integrate GPS-based tracking and prioritization protocols, they typically operate within partially siloed architectures and do not fully incorporate real-time hospital capacity information or structured patient-side digital inputs (Kim & Lee, 2020; Abbasi et al., 2020). As a result, coordination remains predominantly dispatcher-driven and stage-based rather than dynamically synchronized across stakeholders.

The expansion of smartphones and mobile connectivity enabled the development of mHealth applications aimed at facilitating direct patient or bystander reporting. Early systems emphasized geolocation transmission and emergency alerts (Free et al., 2013; Koehler et al., 2015). More advanced applications incorporated structured symptom questionnaires, multimedia uploads, and preliminary triage logic (Mahmood et al., 2019; Roncero et al., 2020). Empirical studies suggest that such applications may reduce call handling delays and improve location accuracy (Brooks et al., 2016; Gabella et al., 2024). However, many mHealth platforms operate independently from dispatch infrastructures and hospital information systems (Elsner et al., 2020; Silva et al., 2021). While they enhance incident initiation and patient engagement, they rarely provide comprehensive integration across dispatch allocation, routing optimization, and hospital preparedness processes. Advancements in GPS, GIS, and algorithmic optimization significantly improved ambulance fleet management and route efficiency. Real-time tracking systems allow dispatch centers to assign the nearest available unit and dynamically adjust routes based on traffic conditions (Aringhieri et al., 2017; Jia et al., 2017). Predictive positioning models further enhance coverage by forecasting demand hotspots (Jagtenberg et al., 2017; Devenci et al., 2020). Despite these operational gains, routing-focused systems generally prioritize transportation efficiency rather than system-wide coordination. Most models do not incorporate structured patient data, hospital capacity metrics, or cross-domain decision logic into allocation algorithms (Kim & Lee, 2020; Abbas et al., 2021). Consequently, optimization remains localized to the transportation layer rather than embedded within an integrated emergency ecosystem. The distinctions between these existing approaches and the proposed coordination architecture are summarized in Table 1.

Table 1. Comparison of Existing Emergency Response Approaches and the Proposed Framework

Aspect	CAD Systems	mHealth Apps	Routing Systems	Proposed Framework
Primary Focus	Call handling and dispatch management	Patient alerting and reporting	Route optimization	End-to-end coordination
Incident Capture	Telephone communication	Mobile application input	Not primary function	Structured digital reporting
Ambulance Allocation	Dispatcher controlled assignment	Managed outside application	Distance and traffic optimization	Capability and status-aware allocation
Hospital Integration	Limited capacity visibility	No direct integration	Minimal integration	Real time capacity matching
Decision Structure	Sequential workflow	Limited emergency support	Transportation optimization	Parallelized coordination logic
System Integration	Dispatch oriented architecture	Independent mobile tool	Logistics optimization module	Unified Patient-EMS-hospital architecture

As shown in Table 1, existing emergency technologies primarily optimize isolated components of the response chain, whereas the proposed framework integrates patient reporting, dispatch coordination, routing optimization, and hospital capacity matching within a unified, real-time architecture.

Hospital preparedness plays a crucial role in determining treatment delays following patient arrival. Structured pre-arrival notification systems and electronic patient care records (ePCRs) have improved communication between EMS crews and emergency departments (Saberri et al., 2017; Wang et al., 2019). Telemedicine-enabled EMS systems further support remote supervision and early activation of hospital pathways (Silva et al., 2015;

Kim et al., 2018). However, interoperability challenges between EMS platforms and hospital electronic health records continue to limit seamless information exchange (Vest et al., 2019; Mears et al., 2020). In many settings, hospital coordination remains partially manual and dependent on voice-based confirmation (Wong et al., 2020; Balamurugan et al., 2021). Real-time hospital capacity visibility is therefore inconsistently integrated into dispatch decision-making.

Across these domains, prior research demonstrates meaningful advancements in individual components of emergency response systems. CAD platforms enhance dispatch administration; mHealth applications improve incident reporting; routing systems optimize travel efficiency; and hospital notification mechanisms strengthen pre-arrival preparation. However, these technological developments have largely evolved in parallel rather than within a unified, end-to-end coordination architecture. Persistent challenges include fragmented system infrastructures, delayed cross-stakeholder communication, limited integration of hospital capacity data into dispatch logic, and insufficient support for parallelized, real-time decision-making from incident detection to definitive care (Andreassen et al., 2020; Munagandla et al., 2024). Notably, few studies propose a comprehensive, mobile-centric, multi-agent framework that simultaneously integrates patient-side reporting, dispatch allocation, routing optimization, and hospital readiness within a synchronized, event-driven ecosystem. These limitations indicate the need for a systematically designed, integrated coordination architecture capable of patient reporting, dispatch allocation, routing optimization, and hospital readiness within a single real-time framework.

3. METHODOLOGY

This section presents the methodology and system-level design of the proposed integrated mobile framework for reducing emergency response delays through real-time, multi-agent coordination supported by AI-assisted decision support. The framework is designed as a service oriented, cloud backed mobile system inspired by advances in mobile emergency systems, Internet of Emergency Services (IoES), and multi-agent architectures for emergency response and decision support (Damaševičius et al., 2023; Nanda et al., 2020; Surya et al., 2025; Li et al., 2025; Han & Choi, 2024). The methodology proceeds by first describing the overall system architecture of the proposed framework, outlining its layered, cloud backed organization and the role of mobile clients, backend coordination services, and data management components(Figure 1). It then presents the multi-agent architectural design, explaining how key stakeholders in emergency response are modeled as autonomous yet coordinated agents with clearly defined responsibilities and interaction patterns(Table 2). The operational workflow of the system is subsequently detailed to illustrate how emergency events are handled from initial detection through dispatch, transport, and hospital handover. This section further explains the real-time coordination mechanisms that enable continuous location tracking, priority aware routing, and dynamic hospital selection based on live system conditions. Finally, the implementation architecture is discussed, highlighting how mobile applications, cloud services, and AI-assisted decision support components can be integrated to support scalable, real-time emergency response coordination.

3.1 Overall System Architecture

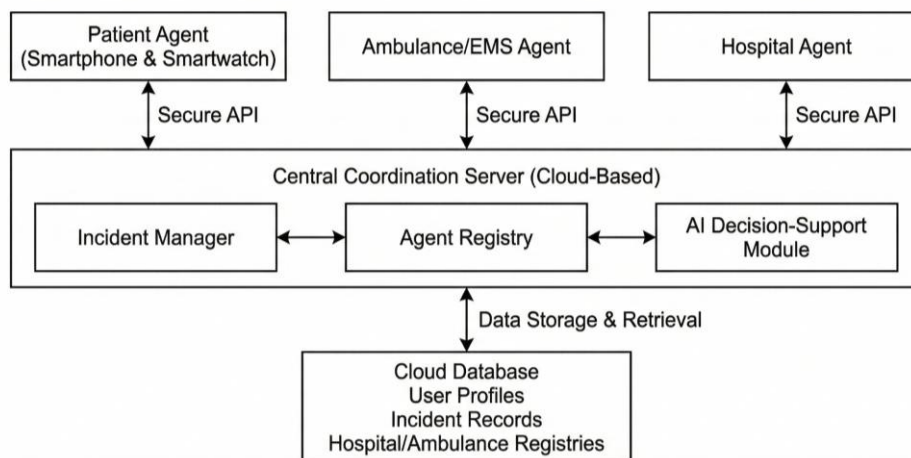


Figure 1. Overall architecture of the proposed emergency response system

The proposed framework is designed as a layered, mobile centric architecture connecting five core agents: Patient, Ambulance/EMS, Hospital, Central Coordination Server through a cloud hosted coordination and data integration backbone(Figure 1). This design is aligned with prior work on mobile and cloud based emergency management and health systems that emphasize scalability, low latency, and real-time data exchange (Nanda et

al., 2020; Munagandla et al., 2024; Mustafa et al., 2025). At the edge, end users interact through Android/iOS mobile applications or wearable interfaces. The Patient Agent runs on the user’s smartphone or smartwatch, capturing location and emergency metadata and transmitting alerts to the backend. Ambulance/EMS personnel and hospital staff use role specific mobile or web clients that provide dashboards for incident lists, routing instructions, and capacity views, consistent with role based coordination patterns in recent mobile health and emergency platforms (Shelar et al., 2025; Medilocker, 2025).

The Central Coordination Server constitutes the core backend layer and is hosted in a cloud environment, supporting horizontal scaling and high availability. It exposes RESTful and streaming APIs to all agents, maintains a real-time event for incident and status updates, and integrates a conceptual AI decision-support layer for incident prioritization and resource allocation. Similar to IoES and real-time data integration frameworks, this layer aggregates geospatial, clinical, and operational data to support coordinated decision making (Damaševičius et al., 2023; Munagandla et al., 2024). A cloud database cluster stores user profiles, historical incidents, ambulance and hospital registries, and configuration rules. Caching mechanisms are designed to accelerate access to frequently used data, following cross layer mobile disaster systems that are used in memory stores for performance in time critical scenarios (Mustafa et al., 2025). Communication between mobile clients and the backend uses secure HTTPS and, for future extensions, SMS gateways for low connectivity environments.

3.2 Multi-Agent Architectural Design

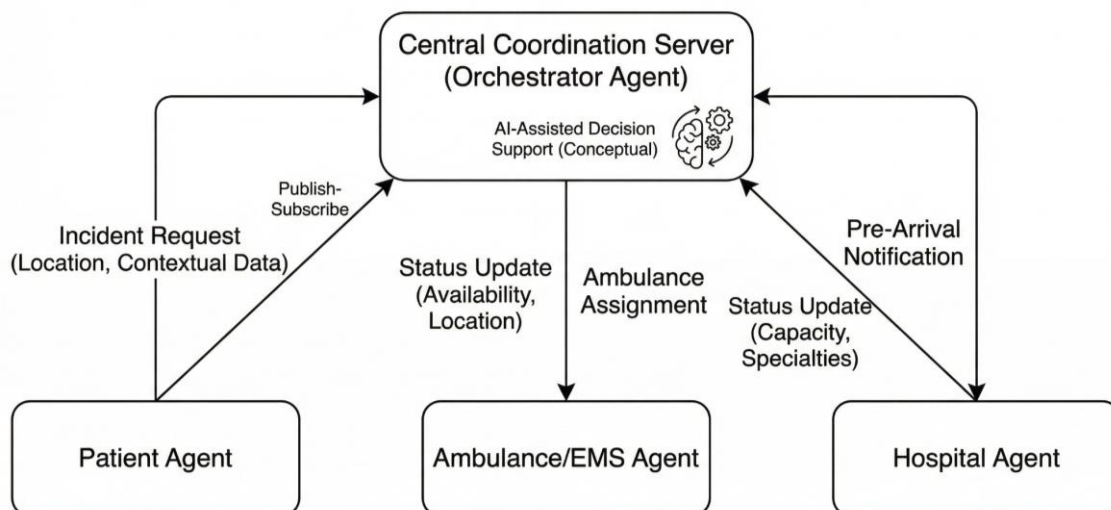


Figure 2. Conceptual multi-agent emergency healthcare coordination framework

The framework is structured as a multi-agent system in which each agent represents a specific stakeholder role and behavior, following recent work on multi-agent emergency dispatch and clinical decision support (Li et al., 2025; Han & Choi, 2024). As shown in Figure 2, the proposed architecture models patients, ambulance and EMS units, hospitals, and a central orchestration component as interacting agents coordinated through event driven communication. Each agent operates autonomously within its role while continuously exchanging structured information through the Central Coordination Server, enabling coordinated decision making across the emergency response lifecycle. The primary roles, responsibilities, and data handled by each agent are summarized in Table 2.

Table 2. Roles and responsibilities of agents in emergency coordination framework

Agent	Primary Role	Key Responsibilities	Data Handled
Patient Agent	Incident initiation	Submits location and incident details; receives status updates and guidance	Location, incident metadata
Ambulance/EMS Agent	Field response coordination	Shares status and location; receives dispatch and routing instructions	Vehicle status, location
Hospital Agent	Capacity and readiness reporting	Publishes availability; responds to pre-arrival notifications	Bed and specialty status
Orchestrator Agent	System wide coordination	Prioritizes incidents; assigns resources; synchronizes agent communication	Aggregated system data

The Patient Agent runs on the end user's smartphone or a bystander's device and is the primary entry point into the system. It guides the user through a simple interface to capture essential data: current GPS coordinates, basic demographics (if available), emergency category (e.g., trauma, cardiac, obstetric), and optional contextual inputs such as photos, short text, or recorded voice describing the situation. Upon activation, the agent automatically time stamps and packages this information into a structured incident report, then transmits it to the Orchestrator Agent via secure APIs as illustrated in (Figure 2). The design mirrors mobile emergency and prehospital assessment tools that rely on geolocation and minimal but structured input to enable rapid classification (Masoumian Hosseini et al., 2023; Singhal, 2025). Once the incident is processed, the Patient Agent maintains a feedback channel: it displays confirmation of receipt, the assigned priority level if exposed to the user, the identification of the responding ambulance, ETA, and the selected destination hospital. It can also surface safety instructions (e.g., stay at location, move to safe area), aligning with AI assisted emergency guidance and symptom checker concepts in emergency care (Kachman et al., 2024; Elfahim et al., 2025). This bidirectional interaction is intended to reduce caller anxiety and support better self management while professional help is on the way.

The Ambulance/EMS Agent represents each emergency vehicle and crew as a continuously connected, context aware node. It runs on an in vehicle tablet or crew smartphone and periodically publishes the vehicle's GPS position, operational status (available, en route, at scene, transporting, unavailable), and key capabilities (e.g., basic vs. advanced life support, presence of specific equipment such as defibrillators or ventilators). Similar real-time based tracking is widely used in intelligent ambulance placement and routing systems (Nahata et al., 2024; Singhal, 2025). The agent subscribes to incident assignment topics from the Orchestrator Agent. When an assignment is received, it fetches an optimized route to the incident site and then to the recommended hospital, taking into account real-time traffic and predicted travel times (Chen et al., 2025; Munagandla et al., 2024). During the mission, the Ambulance/EMS Agent streams status transitions and may optionally upload vital signs or brief clinical summaries gathered in the field. This continuous data exchange enables dynamic fleet management, re-routing, and possible reallocation of units as system conditions evolve, consistent with multi-agent prehospital management and AI-supported EMS dispatch literature (Safdari et al., 2017; Elfahim et al., 2025).

The Hospital Agent abstracts each hospital as an intelligent resource node that provides and updates capacity and service information in real time. It interfaces with local hospital information systems where available, or with a dedicated web dashboard, to maintain: emergency department occupancy, available monitored and ICU beds, specialty readiness (e.g., trauma team, stroke team), and estimated waiting times (Munagandla et al., 2024; González Armengol, 2025). When notified of an incoming patient, the Hospital Agent receives a structured pre-arrival summary from the Orchestrator Agent, including location, suspected condition, initial clinical urgency, and ETA. Based on current capacity and clinical suitability, it responds with an acceptance or redirection decision. This mirrors real-time hospital availability and patient distribution approaches that have been shown to reduce ED crowding and improve access resilience (Golazad et al., 2024; Ajmi et al., 2024). For accepted cases, the Hospital Agent can trigger local workflows alerting emergency intake staff, preparing critical care treatment areas, or reserving imaging slots similar to resource orchestration agents used to optimize emergency department processes (Dhiviya & Nelson, 2021; Ouda et al., 2023). These capabilities support smoother handovers and faster initiation of definitive care.

The Orchestrator Agent resides in the Central Coordination Server and acts as the global coordinator of all other agents. It maintains a unified, real-time view of active incidents from Patient Agents, resource states from Ambulance/EMS Agents, and capacity data from Hospital Agents, in line with situation aware multi-agent emergency platforms (Sako et al., 2024; Munagandla et al., 2024). On receiving a new incident, the Orchestrator Agent validates the data, invokes the AI decision support layer to estimate severity and priority, and then executes two core allocation tasks: (1) selecting the most appropriate ambulance, based on distance, traffic, current workload, and vehicle capabilities, and (2) recommending a suitable destination hospital, using proximity, real-time bed availability, and specialty requirements (Nahata et al., 2024; Elfahim et al., 2025). This reflects multi-agent optimization and coordinated scheduling strategies used in hospital and EMS settings (Dhiviya & Nelson, 2021; Ajmi et al., 2024). The Orchestrator Agent then issues synchronized notifications: assignment orders to the selected ambulance, pre-arrival alerts to the chosen hospital, and feedback to the Patient Agent. It continuously monitors streaming updates and can revise assignments in exceptional cases (e.g., sudden hospital closure, ambulance breakdown). This role parallels orchestrators in multi-agent dispatch simulators and next-generation multi-agent AI systems in healthcare that coordinate multiple specialized agents under real-time constraints (Borkowski & Ben-Ari, 2025; Naim et al., 2021). By including decision support at this orchestration layer, the framework aims to transform fragmented, manual dispatch operations into a data

driven, system level coordination process designed to reduce overall emergency response delays. Agents communicate over asynchronous, event driven channels via the Central Coordination Server using a publish–subscribe model. Each agent publishes state updates and subscribes to topics relevant to its role (e.g., “incident_assigned”, “hospital_status_update”). Decision making is centralized at the orchestration layer but is informed by real-time agent data, enabling coordinated, multi-party workflows rather than isolated, point-to-point interactions (Li et al., 2025; Han & Choi, 2024).

3.3 Workflow of the Proposed System

The operational workflow of the proposed framework is designed to cover the lifecycle of an emergency event from detection to hospital handover. As illustrated in Figure. 3, the workflow progresses from emergency initiation through coordinated agent interactions to hospital handover and incident closure.

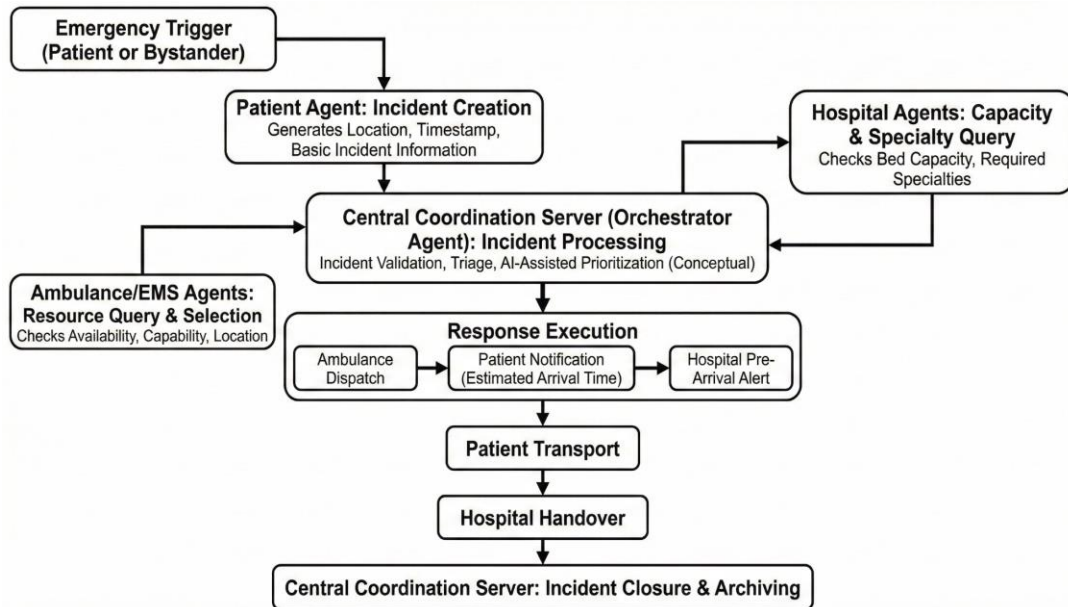


Figure 3. Workflow of the proposed multi-agent emergency healthcare coordination framework.

An emergency can be triggered manually through the mobile app by a patient or bystander, or automatically via connected devices such as wearables and in vehicle crash detection systems, similar to mobile accident detection and IoT-based alerting solutions that use accelerometers, gyroscopes, and GPS to infer high impact events (Satish & Sagar, 2025; Malini et al., 2025; Priyadharshini et al., 2023). The Patient Agent includes GPS coordinates, timestamp, incident category (e.g., road traffic injury, cardiac, obstetric), and optional multimedia (photos, short video, text/voice notes), then creates an encrypted incident request to the Central Coordination Server. Automated triggers help overcome situations where victims are unconscious or alone, a known limitation of purely manual reporting (Satish & Sagar, 2025; J. M. et al., 2024). When the Central Coordination Server receives an incident, it first validates completeness and consistency of the data, then invokes the AI decision-support layer. Using rule based logic and machine learning models by AI driven emergency assessment, it estimates severity and required resources from features such as reported symptoms, contextual descriptors, time of day, and location type (e.g., highway, school) (Da’costa et al., 2025; El Arab & Al Moosa, 2025; Raita et al., 2019). A composite priority score is produced, reflecting both clinical risk and operational urgency. Similar models have shown superior performance to conventional priority scoring systems, improving identification of patients with critical conditions and resource allocation (Raita et al., 2019; Phimphisan et al., 2025).

Given the priority score and geospatial context, the Orchestrator Agent queries the live registry of Ambulance/EMS Agents. It filters candidates by availability, distance, and required capabilities (e.g., ALS, trauma ready), then runs a matching algorithm to select the ambulance expected to minimize response and transport times, in line with multi-agent routing and simulation optimization approaches that reduce delays and improve global system performance (Amr et al., 2021; Aboueljinnane & Frichi, 2022; Olave-Rojas & Nickel, 2021). In parallel, the orchestrator requests real-time status updates from Hospital Agents within a configurable radius, evaluating ED crowding, bed and ICU availability, and specialty capacity. Hospital choice balances travel time with the probability the facility can deliver definitive care on arrival, similar to HIS interoperable recommendation systems and game theoretic hospital selection models in city scale dispatching (Amr et al., 2021; Fu et al., 2025).

Once resources are selected, the Central Coordination Server issues synchronized instructions. The chosen Ambulance/EMS Agent receives an assignment package containing incident location, optimized navigation route using traffic aware models, and preliminary clinical information to guide preparation en route (Amr et al., 2021; Olave-Rojas & Nickel, 2021). The Patient Agent is notified with confirmation, ambulance identifiers, and dynamically updated estimated time of arrival, echoing best practices in mobile emergency apps and accident alert platforms that emphasize user reassurance and transparency (Malini et al., 2025; Soni et al., 2023). Simultaneously, the designated Hospital Agent receives a structured pre-arrival notification (Case assessment level, suspected diagnosis, ETA, key risks) and can pre-activate relevant teams or, if capacity suddenly changes, issue a deferral, prompting re-allocation (Amr et al., 2021; Aboueljinane & Frichi, 2022). Throughout the episode, status transitions from Ambulance/EMS Agents (en route, on scene, transporting, arriving) are streamed to the backend and propagated selectively to Patient and Hospital Agents. This continuous feedback loop supports shared situational awareness and enables dynamic replanning, resembling integrated EMS platforms and live video/telemetry enhanced dispatch systems now being trialed for pre-arrival decision support (Munro et al., 2025; Magnusson et al., 2024; Gude et al., 2025). Handover and Closure, upon arrival at the hospital, the Ambulance/EMS Agent updates the incident status to “handover complete,” and the Hospital Agent confirms reception. The Central Coordination Server archives the full incident timeline for subsequent analysis, quality improvement, and model retraining.

3.4 Real-Time Coordination Mechanism

The real-time coordination capabilities of the framework are organized around three core mechanisms: continuous location tracking, priority based routing, and hospital availability matching. Together, these mechanisms enable the Central Coordination Server to maintain a unified, up-to-date operational picture of all Patient, Ambulance/EMS, and Hospital Agents, and to make rapid decisions about dispatch, routing, and destination selection in a way that reduces response times and avoids overloading specific facilities (Kim et al., 2025; Nahata et al., 2024; Singhal, 2025). All mobile agents, including both Patient Agents (running on patient or bystander smartphones) and Ambulance/EMS Agents (running on crew tablets or in vehicle units), periodically transmit GPS coordinates and timestamps to the Central Coordination Server. This design is aligned with mobile emergency and Internet of Emergency Services (IoES) systems that rely on continuous positioning and geolocation APIs to support rapid unit selection and dispatch (Damaševičius et al., 2023; Shelar et al., 2025; Singhal, 2025). The server maintains an in memory geospatial map of active incident locations, ambulance positions, and hospital coordinates. Using this live map, the system performs proximity calculations and estimates travel times with the aid of mapping APIs and traffic-aware routing services, which have been shown to improve dispatch and routing performance in GPS based and ML enhanced emergency response platforms (Nahata et al., 2024; Selvan et al., 2025). These capabilities enable the orchestrator to quickly identify the nearest suitable ambulance, predict estimated time of arrival (ETA), and continuously update all relevant agents as vehicles move or conditions change (Kim et al., 2025; Singhal, 2025).

The AI decision support layer incorporates a routing engine that computes optimal paths from ambulance to incident and from incident to hospital. These paths are conditioned on the computed priority level, current and predicted road network conditions, and the incident type. For high priority or time critical cases, the routing engine enforces stricter constraints on allowable delays and may select routes that minimize travel time even when they increase total distance, following approaches proposed in smart emergency and rescue routing systems and mixed integer routing optimization under complex traffic (Luan & Jiang, 2024; Vishwakarma et al., 2025). In more advanced scenario, the routing engine can integrate with intelligent traffic management and signal preemption frameworks (e.g., connected vehicle and IoT-based priority control) to clear intersections or dynamically adjust signal phases to favor Ambulance/EMS Agents, similar to AI driven traffic signal control and GPS IoT schemes that significantly reduce emergency vehicle delays (Dai et al., 2025; Prajwal et al., 2024; Alruwaili et al., 2025).

Routing instructions are pushed to Ambulance/EMS Agents in near real time; as congestion, incidents, or road closures appear, the engine recalculates new routes and updates the navigation guidance, to CNN based dynamic route optimization, Google Maps integrated EMS routing (Selvan et al., 2025; Nahata et al., 2024).

Each Hospital Agent periodically reports structured capacity metrics to the Central Coordination Server, including emergency department bed occupancy, ICU bed availability, and current queue length, consistent with real-time integration frameworks and dispatch recommendation systems that forecast bed availability for emergency medical services (Munagandla et al., 2024; Xu et al., 2024; Kim et al., 2025). These updates may also include information on the availability of specific services such as operating rooms, ventilators, or specialized teams, reflecting the “service availability” concept used in ambulance routing and EMS selection

optimization (Almalki et al., 2025). The orchestration layer combines these capacity metrics with geospatial factors (distance and predicted transport time from the incident) and clinical requirements (e.g., trauma center, stroke unit, pediatric ICU) to rank candidate hospitals and choose the most appropriate destination. This multi criteria selection engine seeks to balance rapid access to care with clinical suitability, aiming to reduce diversion, avoid secondary transfers, and distribute patient load more evenly across the network, in line with simulation based dispatch recommendation systems and AI supported prehospital platforms that match patients to hospitals based on both travel time and real-time resource status (Kim et al., 2025; Xu et al., 2024; Almalki et al., 2025).

3.5 System Implementation Architecture

The proposed framework is designed around four conceptual implementation layers: mobile application layer, backend coordination layer, cloud/database layer, and AI decision-support layer. This layering follows best practices in mobile cloud computing for emergency management and real-time health data integration (Nanda et al., 2020; Munagandla et al., 2024). The mobile layer consists of native or cross platform applications for patients, EMS responders, and hospital staff. Core functionalities include emergency alert initiation, live map views, incident status updates, and simple forms for entering clinical or contextual data. The design emphasizes usability in high stress situations, drawing from user centered emergency and disaster apps (Mustafa et al., 2025; Satish & Sagar, 2025).

The backend coordination layer implements the Central Coordination Server and agent APIs. It exposes REST/JSON endpoints and real-time communication channels for bidirectional event streaming. The agent registry, state management, incident lifecycle engine, and notification services reside at this layer. The architecture is intended to be stateless at the API tier and stateful at the service tier, facilitating elasticity and fault tolerance, similar to scalable emergency and IoES platforms (Damaševičius et al., 2023; Zhang et al., 2025). The cloud layer provides storage and data management, using a combination of relational databases for transactional data (users, incidents, assignments) and NoSQL stores for logs and telemetry. Caching systems are designed to accelerate high-frequency queries (e.g., active incidents, nearest ambulances), following secure, cross layer emergency management architectures (Mustafa et al., 2025). Backup and replication strategies aim to ensure high availability and data durability.

The AI layer is designed as a modular service integrated with the backend coordination layer. It includes models for incident prioritization, ambulance and hospital selection, and, potentially, predictive forecasting of demand hotspots, drawing on experience from AI-driven incident assessment, multi-agent dispatch, and real-time emergency classification (Han & Choi, 2024; Li et al., 2025; Vishwakarma et al., 2025). The models are intended to operate in an explainable and auditable manner, producing human readable rationales to support dispatcher oversight. Overall, the proposed methodology presents a scalable multi-agent framework in which mobile clients, cloud services, and AI decision-support components are structurally aligned to support real-time, end-to-end coordination across patients, EMS, hospitals, and other emergency services, with the objective of reducing emergency response delays.

4. RESULTS AND DISCUSSION

To evaluate the feasibility and synchronization of the proposed multi-agent coordination framework, a controlled software simulation environment was implemented. The objective of this validation was not to measure clinical outcomes, but to assess agent synchronization behavior, automated allocation logic, communication consistency, and system stability under controlled workload conditions. Simulation validation is widely adopted in emergency dispatch and IoT-enabled coordination research as a necessary step prior to field deployment, particularly when testing distributed architectures involving real-time communication and resource allocation (Friesen et al., 2024; Damaševičius et al., 2023; Munagandla et al., 2024).

The simulation prototype implemented the four primary agents defined in the architectural model: Patient Agent, Ambulance/EMS Agent, Hospital Agent, and Orchestrator Agent. Each agent operated as an independent software module interacting through a centralized coordination server using a publish–subscribe communication pattern. Transactional interactions such as incident creation and allocation confirmation were handled through RESTful APIs over HTTPS, while continuous synchronization of state updates (ambulance status, hospital capacity, and incident transitions) was maintained using WebSocket bidirectional streaming. The selection of WebSocket protocol was justified by its suitability for low-latency, persistent communication required in real-time emergency systems (Basu, 2019; Munagandla et al., 2024). Similar streaming architectures have been used in mobile-enabled emergency dispatch systems to improve situational awareness and reduce coordination delay (Magnusson et al., 2024; Thajijam, 2023). To ensure deterministic and scientifically meaningful evaluation, structured predefined agents were configured within the simulation. Ambulance agents

were assigned fixed operational states (Available or Busy) and capability levels (Advanced Life Support or Basic Life Support). Hospital agents were configured with specialty categories (Trauma, Cardiac, General) and dynamic ICU/bed capacity parameters. Incident classification rules enforced capability allocation constraints. For example, cardiac incidents required assignment to hospitals with cardiac specialty and available ICU capacity. This deterministic matching logic reflects structured resource coordination strategies discussed in emergency response planning and hospital readiness research (Ceferino et al., 2019; Li et al., 2025).

The system was evaluated under three workload scenarios: single-incident execution, batch simulation of 10 concurrent incidents, and moderate testing involving up to 50 simultaneous active incidents. A finite state machine (FSM) controlled incident lifecycle transitions using the sequence: CREATED → DISPATCHED → ENROUTE → ARRIVED → HANDOVER COMPLETE → CLOSED. The FSM enforcement ensured strict synchronization rules: each incident maintained a unique identifier, no ambulance was assigned to more than one active incident, hospital confirmation was mandatory before final closure, and incident status could not transition to COMPLETED without dual acknowledgment from ambulance and hospital agents. Such structured orchestration aligns with multi-agent coordination models emphasized in emergency management and distributed decision support systems (Andreassen et al., 2020; Han & Choi, 2024). Under moderate testing (50 concurrent incidents), the framework maintained full allocation consistency without duplication, race conditions, or state conflicts. All incident assignments satisfied capability constraints, and no synchronization failures or deadlock conditions were observed. The publish–subscribe communication mechanism maintained stable bidirectional synchronization between agents, confirming the feasibility of event coordination within cloud hosted emergency environments (Damaševičius et al., 2023; Zhang et al., 2025). Performance monitoring tools measured synchronization latency, orchestration computation time, message propagation delay, and allocation success rate. Average agent synchronization latency remained within sub-200 millisecond range, while full orchestration decisions including ambulance matching and hospital selection were consistently completed within one second under moderate load. WebSocket message propagation delay remained below 100 milliseconds in controlled testing conditions. Allocation accuracy and synchronization integrity remained at 100% across simulated scenarios. Communication channels were secured using HTTPS/TLS encryption, consistent with secure mobile healthcare communication frameworks (Siam et al., 2021; Alex et al., 2020).

These findings demonstrate that the proposed multi-agent orchestration architecture validates automated synchronization logic, deterministic allocation enforcement, and low-latency coordination behavior in a controlled software environment. While real-world deployment will introduce additional variability related to network instability and infrastructure constraints, the simulation confirms foundational technical feasibility and provides preliminary evidence that the framework can sustain real-time distributed coordination without structural conflict. Given documented associations between coordination delays and adverse emergency outcomes (Jones et al., 2022), architectural improvements that reduce non-clinical synchronization delays may contribute to improved system efficiency. However, controlled pilot deployment and beta testing remain necessary to evaluate performance under real-world operational conditions.

5. CONCLUSION

This paper presented an integrated mobile multi-agent framework designed to reduce prehospital emergency response delays through real-time orchestration and parallelized coordination. The proposed architecture reconceptualizes emergency response as a synchronized ecosystem of Patient, Ambulance/EMS, Hospital, and Orchestrator agents communicating through event-driven publish–subscribe mechanisms supported by cloud infrastructure. In addition to the architectural specification, a software-based simulation prototype was implemented to validate the coordination logic, automated allocation rules, and agent synchronization behavior under controlled load conditions.

The simulation confirmed resource matching, strict lifecycle state enforcement, and real-time synchronization feasibility without conflicting assignments. Under moderate simulated load, the framework maintained consistent allocation accuracy, bounded orchestration latency, and stable communication performance, demonstrating the technical viability of the coordination model. These results establish foundational feasibility of multi-agent orchestration for emergency response environments prior to real-world deployment. The primary contribution of this work is therefore, a structured system level architectural model for end-to-end emergency coordination, and a validated simulation environment demonstrating logical consistency, synchronization stability, and automated allocation integrity. While the present evaluation does not measure clinical outcomes or real-world operational impact, it provides a technically grounded proof of concept for real-time, cloud based emergency coordination systems.

Recommendations and Future Work

Although the simulation results demonstrate synchronization feasibility and allocation consistency under controlled conditions, real-world implementation introduces additional operational, infrastructural, and human factors that must be systematically evaluated. Future work will focus on staged beta deployment within a controlled institutional or pilot emergency network to validate performance under live network variability and real operational constraints. Planned next steps include deployment on scalable cloud infrastructure, integration with real-time geolocation services, secure interoperability with hospital information systems, and testing under fluctuating network latency conditions. Additional validation will involve stress testing under high incident surge scenarios, evaluation of fault tolerance during partial network failure, and measurement of real-world synchronization delay and dispatch time improvements. Further research should also assess usability, dispatcher oversight integration, governance mechanisms for AI-assisted decision support, and compliance with healthcare data security standards. Clinical validation studies will ultimately be required to determine whether structural coordination improvements translate into measurable reductions in response time, emergency department crowding, and patient outcome delays. Through progressive simulation refinement, pilot implementation, and empirical evaluation, the proposed framework can transition from validated software prototype to operational emergency response coordination platform.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

REFERENCES

- Friesen, J., Kharel, R., & Delaney, P. (2024). Emergency medical dispatch technologies: Addressing communication challenges and coordinating emergency response in low and middle-income countries.. *Surgery*. <https://doi.org/10.1016/j.surg.2024.02.031>.
- Damaševičius, R., Bačanin, N., & Misra, S. (2023). From Sensors to Safety: Internet of Emergency Services (IoES) for Emergency Response and Disaster Management. *J. Sens. Actuator Networks*, 12, 41. <https://doi.org/10.3390/jsan12030041>.
- Basu, T. (2019). A Wireless Framework for Emergency Health Care System. *International Journal for Research in Applied Science and Engineering Technology*. <https://doi.org/10.22214/ijraset.2019.5608>.
- Al-Khafajiy, M., Kolivand, H., Baker, T., Tully, D., & Waraich, A. (2019). Smart hospital emergency system. *Multimedia Tools and Applications*, 1-25. <https://doi.org/10.1007/s11042-019-7274-4>.
- Li, Z. (2024). Leveraging AI automated emergency response with natural language processing: Enhancing real-time decision making and communication. *Applied and Computational Engineering*. <https://doi.org/10.54254/2755-2721/71/20241629>.
- Ceferino, L., Mitrani-Reiser, J., Kiremidjian, A., Deierlein, G., & Bambaren, C. (2019). Effective plans for hospital system response to earthquake emergencies. *Nature Communications*, 11. <https://doi.org/10.1038/s41467-020-18072-w>.
- Munagandla, V., Pochu, S., Nersu, S., & Kathram, S. (2024). Real-Time Data Integration for Emergency Response in Healthcare Systems. *Journal of AI-Powered Medical Innovations (International online ISSN 3078-1930)*. <https://doi.org/10.60087/japmi.vol.03.issue.01.id.002>.
- Jones, S., Moulton, C., Swift, S., Molyneux, P., Black, S., Mason, N., Oakley, R., & Mann, C. (2022). Association between delays to patient admission from the emergency department and all-cause 30-day mortality. *Emergency Medicine Journal*, 39, 168 - 173. <https://doi.org/10.1136/emered-2021-211572>.
- Fukaguchi, K., Goto, T., Yamamoto, T., & Yamagami, H. (2022). Experimental Implementation of NSER Mobile App for Efficient Real-Time Sharing of Prehospital Patient Information With Emergency Departments: Interrupted Time-Series Analysis. *JMIR Formative Research*, 6. <https://doi.org/10.2196/37301>.
- Alharbi, A., Omaier, H., & Alotaibi, M. (2020). Safety souls mobile application for emergency response system. *International Journal of Data Science*. <https://doi.org/10.1504/ijds.2020.10034172>.
- D, R., D, S., E, B., & Ganesh, K. (2024). Ambulance Tracking and Health Monitoring System. 2024 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS), 1-6. <https://doi.org/10.1109/icpects62210.2024.10780080>.

- Yu, J., Liu, C., Chen, W., Yu, S., Huang, F., Yang, M., Chen, C., & Shih, H. (2021). Impact of the COVID-19 pandemic on emergency medical service response to out-of-hospital cardiac arrests in Taiwan: a retrospective observational study. *Emergency Medicine Journal : EMJ*, 38, 679 - 684. <https://doi.org/10.1136/emmermed-2020-210409>.
- Roncero, A., Marques, G., Sainz-De-Abajo, B., Martín-Rodríguez, F., Del Pozo Vegas, C., Garcia-Zapirain, B., & De La Torre-Díez, I. (2020). Mobile Health Apps for Medical Emergencies: Systematic Review. *JMIR mHealth and uHealth*, 8. <https://doi.org/10.2196/18513>.
- Gabella, J., Gualda, I., Altoé, I., Beltrame, M., Da Silva, P., Costa, D., Paggi, F., Filho, S., Massago, M., De Paulo, L., Bitencourt, M., Joiner, A., Vissoci, J., & De Andrade, L. (2024). Designing, development and validation of an app to reduce the response time of the emergency medical services. *PLOS ONE*, 19. <https://doi.org/10.1371/journal.pone.0299828>.
- Thajijam, C. (2023). A Smart Ambulance With Information System and Decision-Making Process for Enhancing Rescue Efficiency. *IEEE Internet of Things Journal*, 10, 7293-7302. <https://doi.org/10.1109/jiot.2022.3228779>.
- Magnusson, C., Ollis, L., Munro, S., Maben, J., Coe, A., Fitzgerald, O., & Taylor, C. (2024). Video livestreaming from medical emergency callers' smartphones to emergency medical dispatch centres: a scoping review of current uses, opportunities, and challenges. *BMC Emergency Medicine*, 24. <https://doi.org/10.1186/s12873-024-01015-9>.
- Rasouli, H., Esfahani, A., Nobakht, M., Eskandari, M., Mahmoodi, S., Goodarzi, H., & Farajzadeh, M. (2019). Outcomes of Crowding in Emergency Departments; a Systematic Review. *Archives of Academic Emergency Medicine*, 7. <https://doi.org/10.22037/aaem.v7i1.332>.
- Metelmann, B., & Metelmann, C. (2019). Mobile Health Applications in Prehospital Emergency Medicine. *Advances in Healthcare Information Systems and Administration*. <https://doi.org/10.4018/978-1-5225-8021-8.ch006>.
- Teng, C., Poshan, Y., & Liu, L. (2024). A cooperative optimization model and enhanced algorithm for guided strategies in emergency mobile facilities. *Humanities and Social Sciences Communications*, 11, 1-11. <https://doi.org/10.1057/s41599-024-02782-3>.
- Andreassen, N., Borch, O., & Sydnes, A. (2020). Information sharing and emergency response coordination. *Safety Science*, 130, 104895. <https://doi.org/10.1016/j.ssci.2020.104895>.
- Zhang, H., Zhang, R., & Sun, J. (2025). Developing real-time IoT-based public safety alert and emergency response systems. *Scientific Reports*, 15. <https://doi.org/10.1038/s41598-025-13465-7>.
- Li, X., Yu, H., Wang, W., Wu, Y., Zhou, J., Hua, W., Lin, X., Tan, W., Zhu, L., Chen, B., Chen, G., Chen, M., Zhou, Y., Li, Z., Assimes, T., Zhang, Y., Wu, Q., , X., Li, L., & Fan, L. (2025). DispatchMAS: Fusing taxonomy and artificial intelligence agents for emergency medical services. *ArXiv*, abs/2510.21228. <https://doi.org/10.48550/arxiv.2510.21228>.
- DispatchMAS: Fusing taxonomy and artificial intelligence agents for emergency medical services. *ArXiv*, abs/2510.21228. <https://doi.org/10.48550/arxiv.2510.21228>.
- Damaševičius, R., Bačanin, N., & Misra, S. (2023). From Sensors to Safety: Internet of Emergency Services (IoES) for Emergency Response and Disaster Management. *J. Sens. Actuator Networks*, 12, 41. <https://doi.org/10.3390/jsan12030041>.
- Surya, D., Durga, M., Jogineedi, P., Ansari, M., Venkata, H., Lingineni, S., Pasupuleti, N., Chiru, G., & Gopal, H. (2025). Optimizing Emergency Equipment Sharing for Rapid Response and Efficiency. *Journal of Information Systems Engineering and Management*. <https://doi.org/10.52783/jisem.v10i19s.3016>.
- Satish, V., & Sagar, P. (2025). Real-time Accident Detection and Emergency Notification System for Mobile Devices. *International Journal for Research in Applied Science and Engineering Technology*. <https://doi.org/10.22214/ijraset.2025.73438>.
- Munagandla, V., Pochu, S., Nersu, S., & Kathram, S. (2024). Real-Time Data Integration for Emergency Response in Healthcare Systems. *Journal of AI-Powered Medical Innovations (International online ISSN 3078-1930)*. <https://doi.org/10.60087/japmi.vol.03.issue.01.id.002>.

Siam, A., Almaiah, M., Al-Zahrani, A., Elazm, A., Banby, G., El-Shafai, W., El-Samie, F., & El-Bahnasawy, N. (2021). Secure Health Monitoring Communication Systems Based on IoT and Cloud Computing for Medical Emergency Applications. *Computational Intelligence and Neuroscience*, 2021. <https://doi.org/10.1155/2021/8016525>.

Saidi, H., Labraoui, N., Ari, A., Maglaras, L., & Emati, J. (2022). DSMAC: Privacy-Aware Decentralized Self-Management of Data Access Control Based on Blockchain for Health Data. *IEEE Access*, 10, 101011-101028. <https://doi.org/10.1109/access.2022.3207803>.

Vishwakarma, A., Vishwakarma, A., & Chauhan, R. (2025). Reducing Urban Emergency Fatalities: A Holistic AI-Driven Rescue Model. *International Journal For Multidisciplinary Research*. <https://doi.org/10.36948/ijfmr.2025.v07i03.43874>.

B, B., Isravel, D., & Dhas, J. (2025). Medilocker: Transforming Healthcare Management with Role-Based Coordination and Real-Time Emergency Response. 2025 8th International Conference on Trends in Electronics and Informatics (ICOEI), 1201-1207. <https://doi.org/10.1109/icoei65986.2025.11013604>.

Borghoff, U., Bottoni, P., & Pareschi, R. (2025). An organizational theory for multi-agent interactions integrating human agents, LLMs, and specialized AI. *Discover Computing*, 28. <https://doi.org/10.1007/s10791-025-09667-2>.

Mustafa, R., Han, J., Sarkar, N., & Petrova, K. (2025). Secure Cross-Layer Mobile Sensing Framework for Real-Time Disaster Reporting and Visualisation Using a Mobile Application. *Sensors (Basel, Switzerland)*, 25. <https://doi.org/10.3390/s25216766>.

Li, J., Wang, S., Zhang, M., Li, W., Lai, Y., Kang, X., , W., & Liu, Y. (2024). Agent Hospital: A Simulacrum of Hospital with Evolvable Medical Agents. *ArXiv*, abs/2405.02957. <https://doi.org/10.48550/arxiv.2405.02957>.

Shelar, Y., Hojane, A., Patel, S., Hate, A., & Shah, D. (2025). Emergency Services. *International Journal of Advanced Research in Science, Communication and Technology*. <https://doi.org/10.48175/ijarsct-24936>.

Alex, S., Pattathil, D., & Jagalchandran, D. (2020). SPCOR: a secure and privacy-preserving protocol for mobile-healthcare emergency to reap computing opportunities at remote and nearby. *IET Inf. Secur.*, 14, 670-682. <https://doi.org/10.1049/iet-ifs.2020.0086>.

Han, S., & Choi, W. (2024). Development of a Large Language Model-based Multi-Agent Clinical Decision Support System for Korean Triage and Acuity Scale (KTAS)-Based Triage and Treatment Planning in Emergency Departments. *Adv. Artif. Intell. Mach. Learn.*, 5, 3261-3275. <https://doi.org/10.48550/arxiv.2408.07531>.