applied sciences

MDPI

Article

ODIN IVR-Interactive Solution for Emergency Calls Handling

Bogdan-Costel Mocanu [1], Ion-Dorinel Filip [1], Remus-Dan Ungureanu [1], Catalin Negru [1], Mihai Dascalu [1], Stefan-Adrian Toma [2], Titus-Constantin Balan [3,4], Ion Bica [2] and Florin Pop [1,5,*]

1 Computer Science and Engineering Department, Faculty of Automatic Control and Computers, University Politehnica of Bucharest, 060042 Bucharest, Romania
2 Faculty of Information Systems and Cyber Security, "Ferdinand I" Military Technical Academy, 050141 Bucharest, Romania
3 Electronics and Computers Department, Faculty of Electrical Engineering and Computer Science, Transilvania University of Brasov, 500036 Brasov, Romania
4 ATOS Convergence Creators, 500090 Brasov, Romania
5 National Institute for Research & Development in Informatics—ICI Bucharest, 011555 Bucharest, Romania
* Correspondence: florin.pop@upb.ro

Abstract: Human interaction in natural language with computer systems has been a prime focus of research, and the field of conversational agents (including chatbots and Interactive Voice Response (IVR) systems) has evolved significantly since 2009, with a major boost in 2016, especially for industrial solutions. Emergency systems are crucial elements of today's societies that can benefit from the advantages of intelligent human–computer interaction systems. In this paper, we present two solutions for human-to-computer emergency systems with critical deadlines that use a multi-layer FreeSwitch IVR solution and the Botpress chatbot platform. We are the pioneers in Romania who designed and implemented such a solution, which was evaluated in terms of performance and resource management concerning Quality of Service (QoS). Additionally, we assessed our Proof of Concept (PoC) with real data as part of the system for real-time Romanian transcription of speech and recognition of emotional states within emergency calls. Based on our feasibility research, we concluded that the telephony IVR best fits the requirements and specifications of the national 112 system, with the presented PoC ready to be integrated into the Romanian emergency system.

Keywords: emergency system; interactive voice response; intelligent human–computer systems

check for updates

Citation: Mocanu, B.-C.; Filip, I.-D.; Ungureanu, R.-D.; Negru, C.; Dascalu, M.; Toma, S.-A.; Balan, T.-C.; Bica, I.; Pop, F. ODIN IVR-Interactive Solution for Emergency Calls Handling. *Appl. Sci.* **2022**, *12*, 10844. https://doi.org/10.3390/app122110844

Academic Editor: Vito Renò

Received: 20 September 2022
Accepted: 19 October 2022
Published: 26 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Major progress in digital transformation has been noticeable in the last years, especially in human–computer interaction systems, due to emerging technologies such as artificial intelligence (AI), the Internet of Things (IoT), Cloud and Edge computing, and 5G networks. Moreover, digital transformation is highly encouraged by international programs—for instance, Europe's Digital Decade provides digital targets for 2030 to strategically empower businesses and people in a human-centered, sustainable, and more prosperous digital future.

The integration of AI technologies in our daily activities, especially in public services, can increase both the productivity of employees as and the satisfaction of the direct beneficiaries, resulting in better public services at a reduced cost. These intelligent agents perform a variety of jobs ranging from routine repetitive work to sophisticated tasks. A typical example of an intelligent human–computer interaction (HCI) agent is a chatbot [1,2] that interacts and responds when conversed with through text or voice and understands one or more human languages via Natural Language Processing (NLP). Another example of an HCI system is an Intelligent Interactive Voice Response (IVR) system, also known as a voicebot [3,4]. Usually, an IVR enables humans to interact with a computer-operated phone system through voice or dual-tone multi-frequency signaling (DFTM) tones input

via a keypad. IVRs are generally used as a complementary solution for enterprises' contact services because of their versatility and high efficiency [5].

The main areas of business that benefit from the advantages of chatbots/IVR systems are [6] supporting systems, education, health care, cultural heritage, and entertainment. A more challenging use case for intelligent human-interaction systems is an emergency system. In case of emergencies, the affected population needs and expects effective support from the state authorities. The call is taken in the shortest possible time by the nearest PSAP (Public Service Answer Point) and directed to the specialized agencies that intervene to solve the case. Additionally, the location of the caller is crucial (http://www.eena.org/view/en/About112/legislation.html—last accessed on 23 August 2022), which has led to the implementation of certain techniques for locating callers of 112 in the telephone network.

Emergency call centers are often flooded with calls. The nature of these events could be real emergencies in a short period, related, for instance, to natural phenomena; pocket calls; or malicious calls. AI tools could help analyze calls and messages, which would reduce waiting times and save more lives. Waiting times can be reduced by more effective filtering of abusive calls and by providing suggestions to operators, allowing them to make decisions more quickly. There are additional ways in which AI can assist operators. According to studies in Denmark [7], emergency operators fail to identify approximately 25% of cardiac arrest cases and therefore miss the opportunity to provide resuscitation instructions. Each year, more than 350,000 cases of cardiac arrest occur outside the hospital. If resuscitation maneuvers are applied in the first minutes, the survival rate increases substantially. Artificial intelligence could fill in this gap and support the early detection of cardiac arrest.

Based on these assumptions, in 2020, Belgium implemented an IVR solution called the Options Menu (https://www.sos112.be/en/menu/—last accessed on 5 October 2022) for the national 112SOS Emergency System. The Belgium approach includes only two options that redirect the citizens to either the police department or fire brigade and ambulance. The Options Menu system was first tested as a pilot project between 1 October 2018 and 17 January 2019 in the provinces of Namur and West Flanders. The populations of these two provinces used the new system without any problem, and the authorities did not receive any complaints, which led to the expansion of the project nationwide.

Moreover, a similar IVR solution was implemented in the United Kingdom for the Devon and Cornwall Police Department (https://www.devon-cornwall.police.uk/contact/non-emergency-101/what-is-interactive-voice-response-ivr/—last accessed on 7 October 2022). The IVR system helps callers reach each person or department that is best placed to resolve their inquiries using a two-level decision tree.

At first glance, there are no significant issues in this use case, but the most crucial aspect of these systems is the time constraint. Therefore, the implementation of a chatbot/IVR solution is a challenging task, especially in the area of real-time task scheduling. Moreover, the system must guarantee 100% acknowledgments and resolution of calls in a fixed period of time, usually less than a minute. Missed calls can lead to disastrous situations or even death. Another aspect that needs to be taken into consideration is user satisfaction and compliance with the system. For instance, when a person is facing a critical situation or a medical emergency, his/her patience is very low, and he/she is likely to drop the call if faced with endless call redirects. As such, our solution must be implemented with a maximum depth of three layers.

Moreover, the implementation of such a system faces multiple problems related to scheduling theory. First, as mentioned in the previous paragraph, we are dealing with a deadline-oriented system. Second, it is a resource management scheduling system because it optimizes finite resource usage, especially in crowded situations. Additionally, the use of intelligent IVR in emergency systems significantly reduces the impact of non-critical situations, such as a small-impact car accident with no casualties, a fallen tree without any damage after a storm, suspicious sounds, or visual appearances in the night. Therefore,

prioritization of calls can be implemented concerning critical deadline satisfaction and the 100% policy.

The primary contributions of this paper are as follows:

- Presenting a viable solution for an interactive chatbot/IVR for emergencies that is ready to be implemented in the National Emergency System 112 of Romania as part of the ODIN112 project. The objective of the ODIN112 research project is to create an integrated computer system for real-time transcription of speech into text for the Romanian language and to recognize the callers' emotional states in emergency 112 calls. This system is based on state-of-the-art deep learning models and aims to improve the management of 112 emergency call recordings and increase the quality, efficiency, and effectiveness of 112 call-management activities. The focus of this paper is on the initial interaction with the user.
- Implementing two proofs-of-concept (PoCs) using different technologies, namely web-based and telephony-based IVR solutions, and performing a side-by-side comparison arguing for the best alternative given the specific requirements and specifications.
- Performing a detailed evaluation of the telephony IVR solution in terms of efficiency, effectiveness, and satisfaction, while arguing that standardized custom decision trees for interaction represent the most adequate approach for an IVR tackling emerging calls.

Even though this article does not present a major scientific breakthrough, we are the first in Romania to design, implement, and evaluate a chatbot/IVR solution tailored to the emergency system in cooperation with the national authorities that manage the National Emergency System 112. The scientific contributions of this paper consist of the design, implementation, and evaluation of a novel component that can easily be integrated into a national emergency system by considering a decision tree for emergencies that classifies user calls into four major categories: (i) medical, (ii) police department, (iii) fire department, and (iv) non-urgent.

2. Related Work

2.1. Research Papers

The main characteristic of chatbots is that they can mimic human conversation and interact with users, but they are not built only for this. Most implementations are platform-independent and do not require advanced computer science engineering skills. In general, chatbots are integrated with group conversations or are shared just like any other contact, and multiple conversations can be carried forward in parallel.

The first human–computer interaction system could be considered the Turing Test ("Can machines think?") proposed by Alan Turing in 1950. The authors of [8] present a rigorous analysis of Turing's work. Moreover, the first so-called chatbot was developed in 1966 and was called ELIZA. Its purpose was to act as a psychotherapist, returning user utterances in a question form [9]. The implementation of ELIZA was based on simple pattern matching and a template-based response mechanism.

Marzava and Nastasescu [10,11] present a novel approach for the Finite Element Method (FEM) with a great impact on behavior prediction used by intelligent human-interaction systems.

Colby, Weber, and Hilf [12] proposed in 1971 a better version of ELIZA called PARRY. The evolution of chatbots had significant development in 2009 when the chatbot called ALICE [13] won the Loebner Prize, an annual competition in artificial intelligence that awards prizes to the computer programs considered by the judges to be the most human-like.

Other remarkable virtual-intelligent personal assistant systems were developed in the early 2010s, such as Apple Siri (https://www.apple.com/siri—last accessed on 23 August 2022), Microsoft Cortana (https://www.microsoft.com/en-us/cortana—last accessed on 23 August 2022), Amazon Alexa (https://www.digitaltrends.com/home/what-is-amazons-alexa-and-what-can-it-do—last accessed on 23 August 2022), Google Assistant (https://assistant.google.com—last accessed on 23 August 2022), and IBM Watson (https://www.ibm.com/watson—

last accessed on 23 August 2022). According to the authors of [6], there are four essential concepts for chatbot development:

- Pattern Matching—predicated on representative stimulus–response blocks. This type of early chatbot did not have storage for past responses, which could lead to looping conversations [14]. Examples of pattern-matching chatbots include ELIZA and ALICE.
- Artificial Intelligence Markup Language (AIML)—XML-based chatbots that are based on dialogue units called categories, which are composed of user inputs and responses [15]; moreover, the successor of AIML chatbots is based on chat-script. An example of an AIML chatbot might be Mitsuku [16].
- Natural language processing (NLP) and Natural language understanding (NLU)—uses machine learning techniques to extract context and meanings from natural-language user inputs to obtain user intention [17]. This type of data is called 'intent' and is the key feature of NLP/NLU chatbots. Examples of these chatbots may include RASA [18] or BotPress [19].

In terms of experiments performed in Romanian, Nenciu, Corlatescu, and Dascalu [20] presented a novel RASA-based conversational agent for the Romanian language capable of communicating in predefined microworlds. Their solution is based on the DIET classifier [21] and classifies intents and extracts entities with high accuracy for a given microworld. For the first microworld, their results indicated an F1 score of 97%, while they obtained a score of 93% for the second microworld. Boroghina, Corlatescu, and Dascalu [22] presented a chatbot for Romanian capable of storing and retrieving information. Their approach is based on a graph-based non-relational database—Grakn [23]—an open-source knowledge graph representation that provides an excellent fit for systems operating with highly interconnected data.

Adamopoulou and Moussiades [6] presented a general architecture of a chatbot, as shown in Figure 1, that includes: a language-understanding module that handles the input data from the user; dialog management, which handles action execution and information; and a response-generation module that outputs the information for the user.

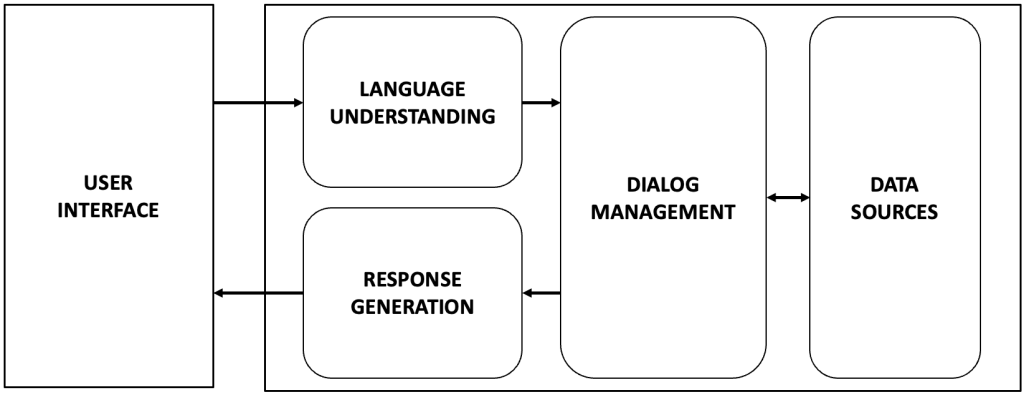

Figure 1. General architecture of a chatbot according to Adamopoulou and Moussiades [6].

Belda-Medina and Calvo-Ferrer [24] presented a study on the level of satisfaction of end-users with interactive human–computer conversational AI-based systems in the educational process. Their research method was based on convenience sampling, and their results were positive. Another approach from educational AI-based chatbots is presented in [25]. Their findings indicate that AI-based chatbot solutions are a viable solution for foreign language learning. In [26], the authors researched, developed, and evaluated an emotion-recognition system for English and Polish texts in the context of a therapeutic chatbot.

2.2. Existing Platforms for Building Conversational Agents

Rasa [18] is an open-source framework for developing conversational agents. It contains all the necessary tools for automatic text processing, message understanding, and conversation maintenance. From the architectural point of view, a Rasa conversational agent is built from a series of modules of different types: segmenting text ("tokenizers"), representing text in vector form ("featurizers"), classifiers for intent, entity extractors, and others. Similar to other frameworks, their components and parameters are chosen according to the particularities of the training data and the final purpose of the agent being built.

There are several important key concepts within a conversational agent developed using this platform: domain, intent, action, and story. The domain defines a microworld in which the conversational assistant operates. Specifically, it contains a string of intents, intent parameters, entities, and actions. As such, it defines the configuration parameters of the agent. The intent dictionary lists all the possible intents a user can have and phrases that describe that intent. These examples may contain parameters, which will then be passed to the action's execution mode. From a string of intentions, actions, and responses, a story can be built, and thus, the agent can maintain context and have a dialogue. The most important part of implementing a Rasa conversational agent is the training stage. At this stage, it is necessary to define all the intentions alongside corresponding examples. If intent refers to entities, then they too must be defined as either rules or entity lists. Optionally, scenarios can be defined for the dialog module.

The Microsoft Bot Framework [27] together with Azure Bot Service are tools for creating, testing, and deploying intelligent bots. The Microsoft Bot Framework includes a modular software development kit (SDK) for building chatbots that understand and process natural language. A major advantage of developing a bot using Microsoft technologies is that it can be integrated with a Microsoft API called Microsoft Cognitive Services. These services are a set of machine learning algorithms. To introduce natural language processing functionality, the Microsoft Bot Framework must be configured to use Microsoft LUIS [28]. Microsoft LUIS is a service used to train chatbots for natural language processing.

The Microsoft Bot Framework offers multiple tools and components for developers, including automatic translation tools, dialog state management, and debugging. The main components of this framework are:

- Bot connector service that uses a REST API and JSON over HTTPS for message exchange;
- Bot Builder SDK for .NET Framework for bot development using the C# programming language and Node.Js.

Botpress [19] is an open-source tool for the visual design of chatbot solutions. Developing a chatbot using Botpress is done by combining visual techniques with writing code. The solution provides a consistent and complex suite of tools for understanding text and automating complex conversations. Botpress was launched in 2015 and is currently one of the most frequently used solutions for creating chatbots according to the authors of [29].

Botpress is a framework distinguished by its execution speed and underlying mechanisms for understanding natural language. The framework enables variable extraction and intent detection and integrates with various communication channels (Web, WhatsApp, Slack, etc.). SWAT analysis of the Botpress framework [30] indicates that the solution is easy to use and has low development cost. On the other hand, modifying the framework to implement functionality specific to the beneficiary is not trivial. Another strength of Botpress is that the solution is written in TypeScript and works on Linux, Windows, and macOS with an AGPL v3 license.

Botpress supports multiple parallel chatbot instances with multiple channels dedicated to NLP providers (Rasa NLU, DialogFlow from Google, and LUIS from Microsoft) and communication platforms (Facebook Messenger).

The four major components of the Borpress platform are:

- User Interface (Chat)—Composed of the means used for I/O operations;
- Input data analysis—Composed of the channels used to communicate with the user as well as the natural language understanding (NLU) component to determine user intent;
- Logic and Dialog manager—Composed of a dialog manager used to determine the optimal flow of conversation;
- Chat Response Training—Contains two components:
  - Content elements—determines the best terms for building optimal responses;
  - Content rendering—builds the answer using the given terms.

2.3. Existing Romanian Chatbot Solutions

Druid (https://www.druidai.com/conversational-ai-platform—last accessed on 19 September 2022) is a Romanian solution for building AI-based conversational agents. This platform offers a cutting-edge proprietary NLP/NLU engine with high accuracy (i.e., over 95%, as reported by the provider) that enables fast building and deployment of AI virtual assistants that instantly assess intent, understand complex queries, and engage in human-like interactions. Their solution interprets user intent and provides real-time contextual information based on behavior and preferences for an enhanced conversational experience.

Holisun (https://holisun.com/noutati/rolul-unui-robot-de-chat-pentru-universitati—last accessed on 16 September 2022) is a Romanian company with 20 years of experience in the research and development of software solutions for business optimization, education, health, defense, and augmented reality. Among the solutions offered by Holisun is a chatbot platform called AIDA.AI, whose main advantages are, as defined by the provider:

- Intelligent chat that integrates both bots and human operators;
- Automated assistance and support process by defining question-and-answer flows specific to the field of activity;
- Knowledge base created and updated by the beneficiaries according to the needs of their activity;
- Automatic redirection of the dialogue to a human provider when AIDA.AI cannot provide an answer;
- Savings of over 80% of human operator support time.

In Romania, the first university to implement such a solution was the Iuliu Haţieganu University of Medicine and Pharmacy in Cluj-Napoca, starting with the international admissions session in July 2021. A robot called 'Ana' was trained to answer most of the questions coming from the candidates using a consistent database of e-mail texts exchanged between the candidates and University employees. The database was formed in 2020. The robot was implemented on day 66 (of 84) of admission; in the 18 days in which it operated assisting human operators, Ana provided 15,843 responses out of a total of 16,459 (96.3%). The bot successfully answered the repetitive questions; as such, only 809 messages reached university employees through all communication channels.

In addition, the 24/7 availability of chatbots is a huge benefit to candidates who live in areas with different time zones and who are not able to interact directly with a human operator. As the Ana robot works, the machine learning component allows it to tackle increasingly complex questions on its own and depend less and less on humans for its work. Finally, the Holisun solution based on AIDA.AI is not an open-source solution.

3. Method

In this section, we define the minimal requirements and specifications for the emergency IVR solution, and we propose an overarching architecture, followed by two PoCs using different technologies, namely web-based and telephony-based IVR solutions. On the one hand, the web-based approach is suitable for a public, cloud-based scenario for which all users must have an Internet connection, while the telephony-based one is a more

robust approach in terms of coverage and can work even in remote areas where telecom operators do not provide Internet services.

3.1. Requirements and Specifications

The requirements and specifications of an IVR/chatbot solution for emergency systems differ from the requirements of corporate chatbots [31]. On the one hand, critical deadline fitting is the most important feature of such a system. Another aspect that needs to be taken into consideration is 100% call handling: no emergency can be left unsolved, even if the situation is not urgent or it is a fake call. On the other hand, another important aspect of emergency systems is that their resources are limited; therefore, such a system must handle multiple parallel emergency calls with a minimum of computing resources. Furthermore, the system's availability must be considered both in terms of coverage and availability time (24/7). By coverage, we understand the percentage of end users that can use the system. This percentage is strictly related to the network operators' data coverage. Table 1 summarizes the specifications and requirements of the chatbot/IVR solution tailored to emergencies as agreed to by the Romanian institution responsible for tackling emergency calls within the framing of the ODIN 112 project. As can be observed, in terms of functionality, the solution needs to be oriented toward low resource consumption. For instance, the CPU utilization of the system must be less than 70%, and the amount of RAM consumed must be less than 2 GB. The maximum time to treat an emergency is two minutes.

Table 1. Requirements and specifications of the chatbot/IVR solution.

Performance		
Critical deadline	<120 s	Beneficiary requirement
Emergency handling percentage (SLA)	99%	Beneficiary requirement
Hardware resources—RAM consumption	<2 GB	Beneficiary requirement
Hardware resources—CPU utilization	<70%	Beneficiary requirement
Round-trip delay	<100 ms	[32]
Lost rate	<1%	[32]
Jitter	<30 ms	[32]
Robustness to unexpected input	91–93%	[33]
Functionality		
Coverage	95%	Beneficiary requirement
Availability	24/7	Beneficiary requirement
Input method	text	Beneficiary requirement
Output method	audio	Beneficiary requirement
Interprets commands accurately	YES	[34]
Executes requested tasks	YES	[35]
General ease of use	YES	[34]
Humanity		
Transparent to inspection (known chatbot/IVR)	YES	[36]
Does not have to pass the Turing Test	YES	[35]
Satisfaction		
Provides greetings	YES	[37]
Protects and respects privacy	YES	[34]

3.2. Architecture and Design

In this section, we present our solution for a chatbot/IVR that can be used in real-life situations, especially in the emergency system. Our proposal identifies non-urgent situations from the emergency call service to prioritize them to fit the hard deadlines. Therefore, we implement the chatbot/IVR solution in the ODIN112 project.

The solution proposed is a chatbot/IVR module, a component of the ODIN 112 system that could endure during a large-scale emergency following manual activation by an

authorized administrator, with the automatic pickup of calls that cannot theoretically be answered in 2 min and that asks standard prerecorded questions that only admit simple answers. These scenarios take into account the number of logged-in users at the national level, the average processing time of an urgent call of approximately 60 s or 10 s for a non-urgent call, and a percentage distribution between them that can be configured in the area of administration (e.g., 50%–50%). If the call is classified as a non-urgent situation, it is transferred without the urgent label, and the user receives a web form to fill in the information regarding the non-urgent event. The architecture of the chatbot/IVR solution is presented in Figure 2.

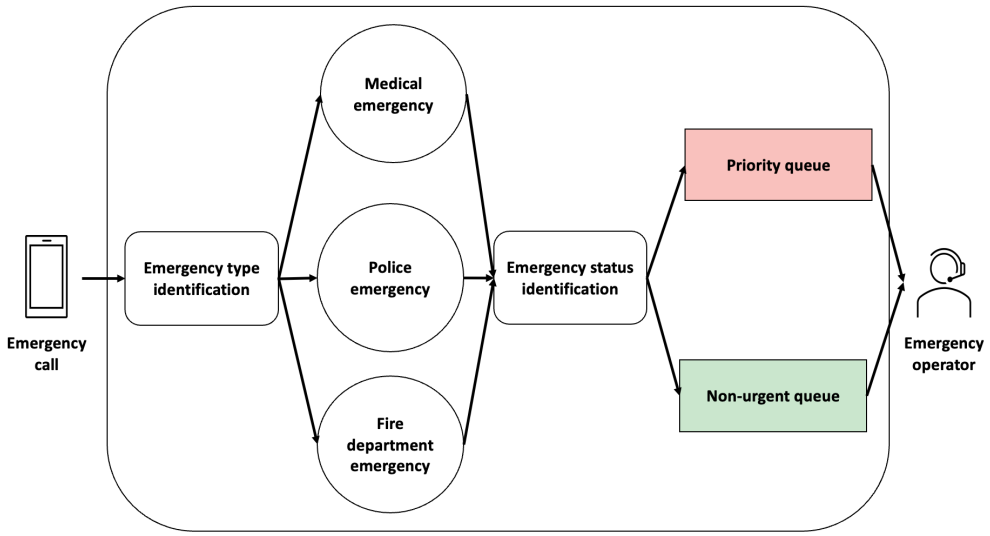

Figure 2. Chatbot/IVR solution business logic flow.

Moreover, in Figure 2, we show the business logic of the chatbot/IVR solution; it contains two components. The first one is a multi-layer module that identifies the emergency type, namely whether it is a police-related, medical, or fire department emergency. Based on beneficiary requirements, all emergency categories were divided into multiple subcategories, thus resulting in a multi-layer approach. The second component identifies whether the situation is urgent or non-urgent. In both cases, the call is handled by an emergency operator.

Thus, we propose a lightweight architecture for a chatbot solution that fits the emergency use case derived from the ODIN112 project.

The chatbot/IVR module is a standalone solution that needs to be integrated into the national emergency system in such a manner that does not affect the quality of the service and does not decrease user satisfaction, especially when users are extremely anxious when facing emergency/critical situations.

Table 2 presents the main categories of emergencies based on beneficiary requirements, along with their emergency statuses: the *RED* status indicates that the situation is urgent and must be treated with high priority; the *GREEN* status indicates a non-urgent situation that can be treated in the order of arrival.

Table 2. Emergency statuses of the situations handled by the chatbot/IVR module.

Type	Name	Emergency Status
Medical	Cerebral vascular accident (CVA)	RED
	Unconscious person	RED
	Suffocation/Severe respiratory problems	RED
	Limb amputation/Severe hemorrhage	RED
	Other medical situation that puts life in imminent danger	RED
	Other medical situation that does not put life in imminent danger	GREEN
Police	Road accident with victims or with a danger of explosion	RED
	Road accident without victims	GREEN
	Railway accident/Subway accident	RED
	Other type of accident that puts life, the environment, or property in imminent danger	RED
	Other type of accidents that does not put life, the environment, or property in imminent danger	GREEN
	Threat/attack with guns or explosive devices	RED
	Violent act with possible victims	RED
	Deprivation of liberty/kidnapping	RED
	Other police competence situation that puts life, the environment, or property in imminent danger	RED
	Other police competence situation that does not put life, the environment, or property in imminent danger	GREEN
Fire department	Fire/explosion	RED
	Storm/tornado/flood/drowning	RED
	Landslide/earthquake	RED
	Naval/aviation accident	RED
	Industrial/chemical accident	RED
	Mountain/ravine/cave accident	RED
	Other type of accident that puts life, the environment, or property in imminent danger	RED
	Other type of accident that does not put life, the environment, or property in imminent danger	GREEN
	Other fire department competence situation that puts life, the environment, or property in imminent danger	RED
	Other fire department competence situation that does not put life, the environment, or property in imminent danger	GREEN

3.3. Web-Based Chatbot Solution

In general, chatbots are task-oriented software systems that handle requests from a specific predefined microworld [38]. This means that a chatbot solution relies on a specific corpus tailored to a use case. As mentioned, our solution is designed for emergencies; therefore, we define a microworld for three scenarios:

- Police emergencies—handles police-related emergencies such as accidents, deprivation of liberty, and violence;
- Medical emergencies—handles emergency calls for medical conditions that are critical and can cause severe damage or even death, such as cerebral vascular accidents, unconscious persons, suffocation, and severe hemorrhage;
- Fire department emergencies—handles fire department-related emergencies such as fires, explosions, extreme natural phenomena, and accidents in low-access areas.

Table 3 introduces the intents and the actions used for the previous use cases. The created corpus contains 30 sentences on 5 possible intents. The limited number of sentences

is directly correlated with the most frequent scenarios encountered in the training manual for 112 services.

Our first PoC was a web-based solution built on the BotPress framework. We implement a docker-contained chatbot that handled the three previous main situations. In Figure 3, we show an example of a use-case diagram built using the GUI. The web-based solution successfully integrated the Botpress built-in NLU engine for intent recognition from text. The model trained for the NLU engine was prebuilt in the BotPress solution, but it can be enhanced or modified with any custom pre-trained model.

We tested the functionality of the Botpress PoC with three simple testing scenarios for the presented use cases. We conducted a manually cross-checked validation of the intent and the behavior of the chatbot. Experimental testing of the PoC was carried out using 20 simple words/sentences correlated to the use cases. We compared the results of the chatbot with the expected results derived from the training material of 112 operators. The results indicate good performance; all tests were passed successfully. It is worth mentioning that the sample words/sentences were chosen from those configured in the Botpress intent definition. The purpose of this validation was only to test the basic functionality of the platform to consider it a viable solution.

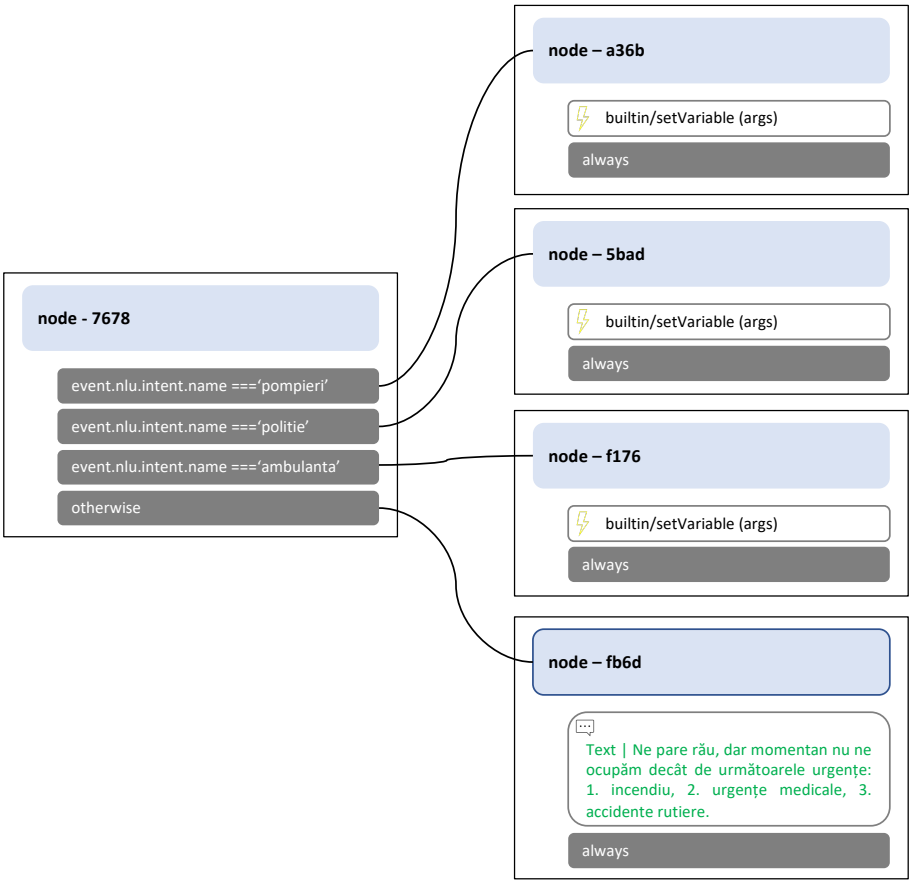

Figure 3. BotPress PoC business logic exemplification.

Table 3. Intent descriptions.

Intent	Description
Greet/Hello	The agent greets the user and asks him/her what the emergency is. Then, the agent tells the user that the chatbot is handling only medication, police, or fire emergencies.
GetEmergencyType	The user text his/her emergency type with words such as "fire", "accident", or "medical emergency".
GetLocation	The agent asks for the location of the user using a sentence consisting of the interrogative pronoun "unde?" (English, "where?") followed by the emergency type.
GetName	The agent asks for the name of the user using a sentence consisting of "Cum vă numiți?" (English, "What is your name?").
GetConclusion	The agent acknowledges the user's emergency using a phrase that contains the name of the user and the conclusion. An example is "Bogdan, trimitem un echipaj de poliție la adresa." (English, "Bogdan, we have sent a police car to the address").

3.4. Telephony IVR Solution

Furthermore, we introduce a second technical solution for the chatbot/IVR feature of the project, as we designed and implemented a telephony-based PoC based on existing open-source solutions of Session Initiation Protocol (SIP). This approach was feasible because it is easy to integrate with the current infrastructure of the National Emergency Systems 112.

Based on our research, we considered that the FreeSwitch [39] open-source software-defined telecommunication stack that best fits the beneficiary requirements and specifications. As mentioned in Section 3.1, the input of the systems must be written. Thus, in telephony systems, the most suitable approach is the DFTM method. The architecture of this solution (Figure 4) consists of several software modules, including:

- DFTM module for user-written input;
- Audio module for system audio output;
- Decision tree for the business logic of the IVR solution.

The decision tree of the proposed solution is built on fuzzy-logic if-then-else statements [40] and covers all the situations presented in Table 2. We modeled the decision tree used for the IVR solution as a tree with 27 leaf nodes, from which 7 are considered non-urgent situations and 20 as urgent situations. We assessed that this approach was one of the most convenient because all states are well-known and the input from the user is straightforward.

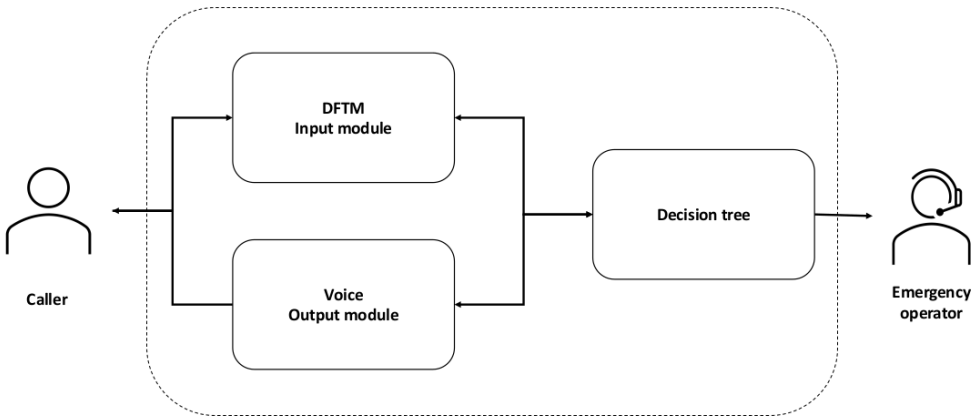

Figure 4. Telephony-based IVR architecture.

Our implementation of the PoC uses the LUA programming language [41]. LUA is an efficient, powerful, lightweight, and embeddable scripting language that integrates well with the FreeSwitch stack.

4. Results

4.1. Performance Analysis of the PoCs

The preliminary evaluation of the proposed solutions consists of performance resource consumption in terms of CPU load and memory SLA (service level agreement) compliance. Therefore, we test both solutions on the same machine (a general-purpose computer with an Intel CPU@3.2 GHz, 12GB RAM, and 512GB SSD).

Our first experiment is resource consumption analysis. The tested scenario involves the same number of messages for the web-based solution and calls for the telephony one. The results indicate that the telephony-based PoC is more efficient than the web-based solution, as shown in Figure 5.

Even though the web-based solution is a scalable trending approach because it uses modern and efficient technologies, it is not satisfactory as the National Emergency System is based on a static infrastructure that is scaled occasionally.

The second considered metric is SLA compliance. As such, we want 100% handling with a maximum 1500 ms delay for both solutions. The obtained results indicate that both solutions perform satisfactorily in terms of SLA compliance, but the telephony-based PoC performs better than the web-based one with a 4.26% boost, as indicated in Figure 6.

To conclude this section, we consider the telephony-based approach the most suitable technical solution for the chatbot/IVR component of the ODIN112 project based on our experimental evaluation and the beneficiary requirements.

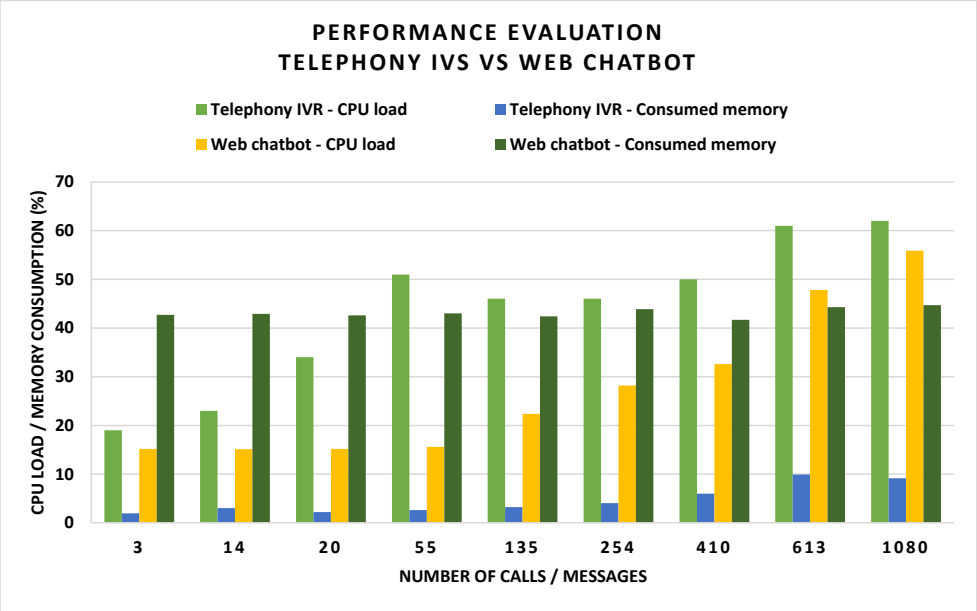

Figure 5. Resource consumption evaluation of the PoCs.

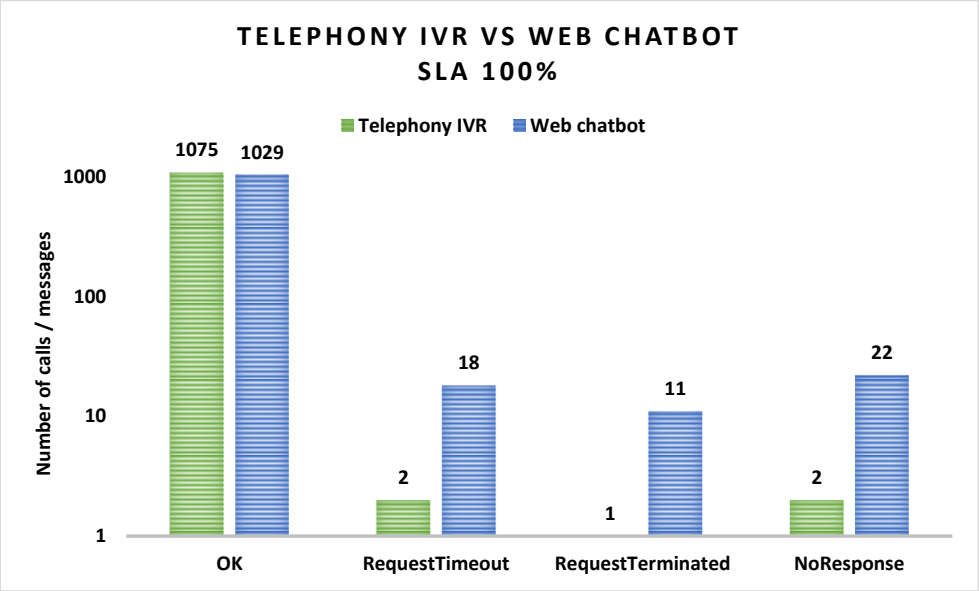

Figure 6. SLA compliance evaluation of the PoCs.

4.2. Experimental Evaluation

Based on the conclusion drawn in the previous section, we extend the experimental evaluation only to the telephony-based PoC.

4.2.1. Efficiency Evaluation

We consider additional metrics that indicate the performance of our solution in terms of concurrent call handling, round-trip delay of the RTCP packets, packet loss, and also the jitter for both the called and the called entity. For the experimental evaluation, we use a specialized software SIP server quality-assessment tool called StarTrinity SIP Tester [42]. The experimental setup consists of a test case of 1080 calls. The total amount of time needed for the experimental evaluation is 20.12 min. Figure 7 shows the evolution of concurrent calls in the 20.12 min timeframe. The results indicate that our solution can handle a large number of calls, including 50 simultaneous calls.

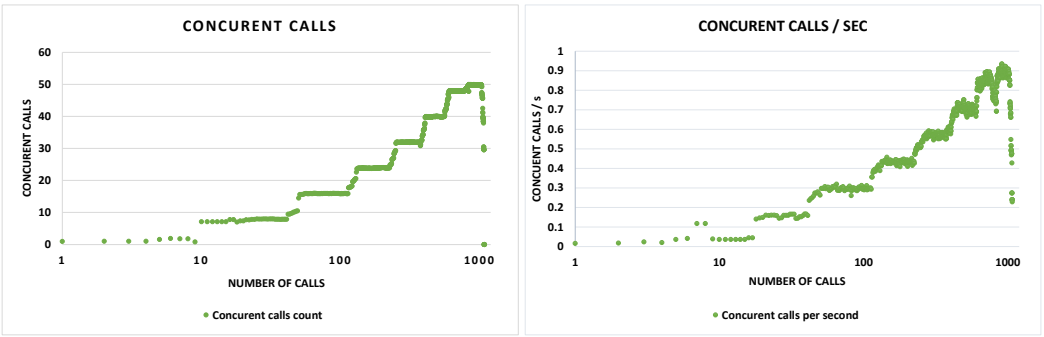

Figure 7. Concurrent calls evaluation: evolution of concurrent calls in 20.12 min (**left**), and evolution of concurrent calls per second (**right**).

The second considered metric is the round-trip delay (RTD) for the RTCP (real-time control protocol) packets. The obtained results presented in Figure 8 show good performance of the PoC, as the average RTP delay was under 20 ms.

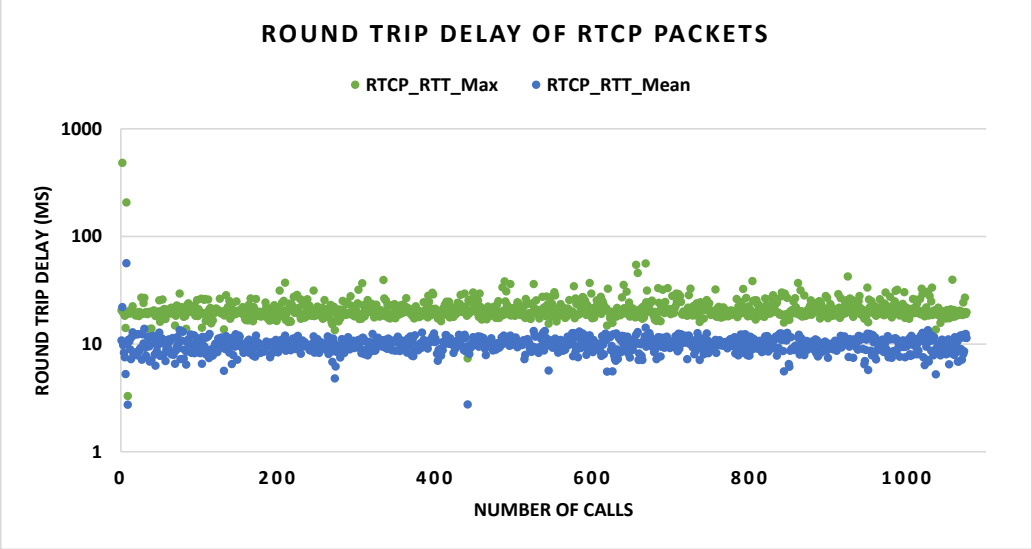

Figure 8. Evaluation of round-trip delay of the RTCP packets.

In correlation with the RTD, we also evaluated the packet loss rate. Taking into consideration the same experimental use case, we obtain an average loss rate of 0.2–0.3%, which denotes good performance of the PoC (see Figure 9).

Finally, the last metric is the jitter both for the caller and the server. The results show values below 1.4 ms, which is an acceptable value for SIP applications (see Figure 10).

In terms of robustness to unexpected inputs, we tested our IVR solution in four scenarios with erroneous inputs; the results indicated that our solution does not crash or lead to unexpected behaviors.

Table 4 summarizes the experimental results and assesses them against the specifications and requirements presented in Section 3. As can be observed, all results satisfy the minimal requirements and specifications, thus arguing for the adequacy of the second proposed solution.

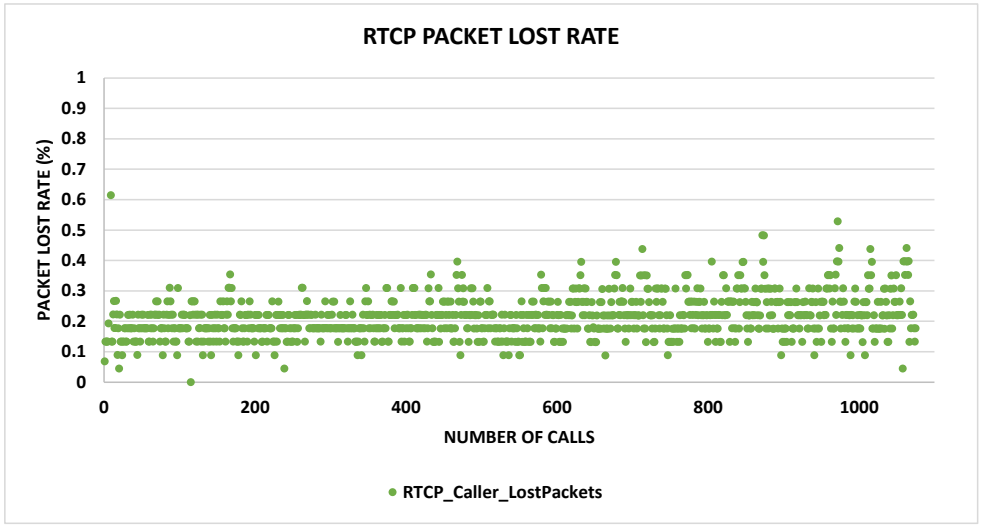

Figure 9. Evaluation of packet loss rate.

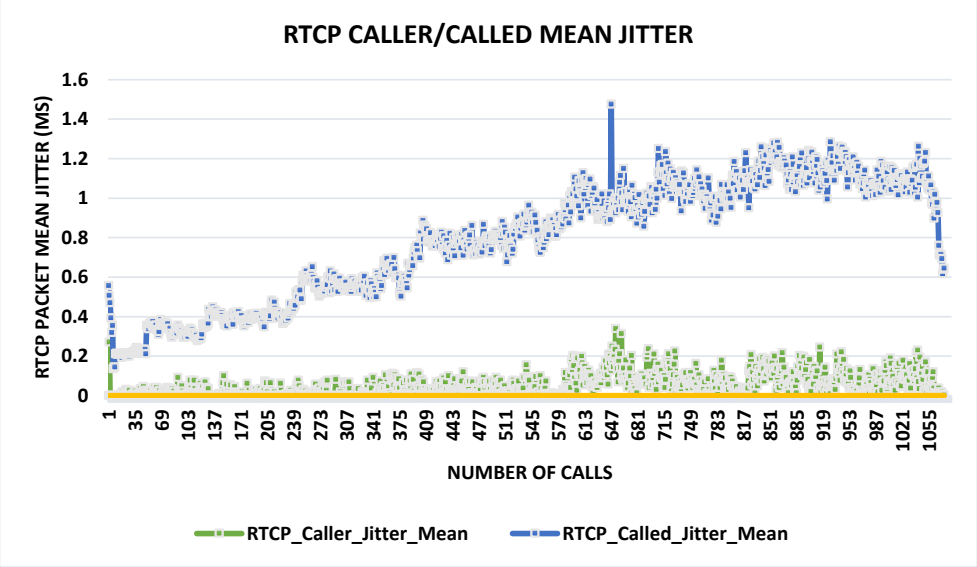

Figure 10. Jitter evaluation for caller RTCP packets (green) and called-entity RTCP packets (blue).

Table 4. Summary of efficiency evaluation.

Metric	Minimal	Evaluated Value
Critical deadline	<120 s	60 s
Emergency handling percentage (SLA)	99.5%	100%
Hardware resources—RAM	<2 GB	1.2 GB
Hardware resources—CPU	<70%	62%
Round-trip delay	<100 ms	20 ms
Loss rate	<1%	0.2–0.3%
Jitter	<30 ms	1.4 s
Robustness to unexpected input	91–93%	100%

4.2.2. Effectiveness Evaluation

In terms of coverage and availability, we evaluate the IVR solution in laboratory environments, and the results indicate the perfect functionality of the system. Taking into consideration that the proposed solution is designed for the Emergency System, it will use the infrastructure of that system, which complies with the specifications. Moreover, we conduct an empirical investigation using 1015 laboratory experimental calls to the IVR solution, of which 235 are related to medical emergencies, 265 are for the police department, 233 are for the fire department, and 282 are related to non-urgent situations. We analyze the responses of the decision tree and compare them with the expected results, as shown in Figure 11. As observed, the obtained results are compliant with the emergency statuses presented in Table 2. The percent of non-urgent situations detected among medical, police, and fire department situations correlates with the number of non-urgent leafs of the decision tree (1 non-urgent leaf from 6 medical emergencies, 3 non-urgent leafs from 10 police emergencies, and 2 non-urgent leafs from 10 fire department emergencies). Therefore, the experimental evaluation indicates the fact that our approach is able to determine both urgent and non-urgent situations, even though the emergency first came from an urgent branch (medical, police, or fire department). On the other hand, the decision tree proves to be a suitable solution; because of its construction, it does not allow the misunderstanding of emergency types.

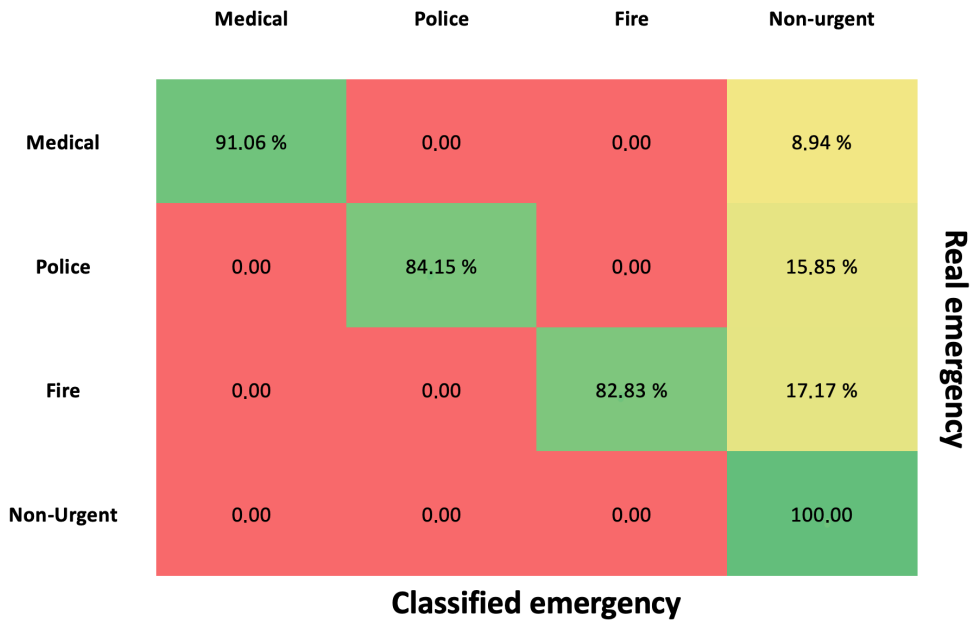

Figure 11. Decision tree evaluation for the empirical laboratory experiments.

We designed the IVR to take simple text input (DFTM) from the caller and offer audio output. We also implemented text output to the log system. Table 5 presents the obtained results and assesses them against the specifications and requirements from Section 3; the results argue for 100% compliance with the requirements.

Concerning the ease of use, the IVR proposal uses simple DFTM input with values from 0 to 5. The IVR does not require multiple key press actions or other complex input schemes.

In addition, we conducted multiple functional testing experiments to assess the functionality and the desired behavior based on the decision tree presented in Table 2. We found that the IVR does not have unexpected behaviors, and it interprets and executes requests as they are given.

As such, an IVR designed for emergency systems is not required to interact humanly with the caller, but rather to efficiently transfer a caller to the emergency operator. Therefore, the IVR does not implement any functionality that could pass the Turing test.

Table 5. Summary of effectiveness evaluation.

Metric	Minimal	Obtained Result
Coverage	95%	100% (in testing environment)
Availability	24/7	24/7 (in testing environment)
Input method	text	text
Output method	audio	audio and text (for log messages)
Interprets commands accurately	YES	YES
Executes requested tasks	YES	YES
General ease of use	YES	YES
Transparent to inspection	YES	FreeSwitch-based
Does not have to pass the Turing Test	YES	YES

4.2.3. Satisfaction Evaluation

In terms of humanity evaluation, we assess the IVR for the greeting requirement (see Table 6), as our solution offers a welcome greeting and an exit phrase. We mitigate the privacy concerns, as the IVR is designed for the Emergency System, and it will be

deployed in a protected/secure environment that complies with specific privacy and security standards.

Table 6. Summary of humanity evaluation.

Metric	Minimal	Obtained Result
Provides greetings	YES	YES
Protects and respects privacy	YES	YES

5. Conclusions and Future Work

Based on the proposed architecture and the specific requirements defined within the ODIN112 project, we developed two tailored solutions for automatic human–computer interaction for the Emergency System, namely a web-based chatbot and a telephony-based IVR, that consider trending technologies and frameworks.

In the first round of evaluation, we made a critical comparison between the two approaches based on both resource consumption and QoS metrics. The results show that the IVR solution is a better choice in terms of resource consumption. Our analysis indicates on average 16% less CPU load for the telephony IVR solution in comparison to the web-based solution. In terms of memory consumption, we conclude that the telephony-based approach has better performance than the web-based chatbot, with an average of 38.45%.

Taking into consideration the results from the preliminary evaluation, the second round of evaluation was focused only on the IVR. In this stage, we assessed several performance metrics for SIP solutions. Overall, the obtained results indicate that our solution is a highly competitive and reliable solution according to the Basic Telephony SIP End-to-End Performance Metrics [32].

In comparison with the Belgium IVR solution Options Menu, our solution offers a more complex and elaborate decision tree for emergencies. The depth of our solution is three times the depth of the Belgium Options Menu. For instance, ODIN IVR is able to treat road accidents separately from subway accidents, while the Options Menu IVR handles both emergencies in the same manner. Additionally, our approach offers the emergency status of each call related to the emergency type and also identifies non-urgent situations correlated to a specific department.

In conclusion, our solution satisfies both the requirements imposed by the beneficiary and the quality metrics. As mentioned, the proposed PoC was also tested with real data as part of the computer system for real-time transcription of speech into text for the Romanian language and recognition of emotional states in emergency 112 calls—the ODIN112 project. In terms of future work, we also intend to evaluate end-user satisfaction in a pilot project within a limited geographical area (e.g., 1–2 counties in Romania) while considering multiple criteria, such as the number of users that finished the call with success (from dialing to emergency operator transfer), degree of trust in the IVR solution, the soundness of the IVR voice, and the degree of acceptance by the end-users. These metrics can be evaluated only in production with real emergencies and people. To the best of our knowledge, satisfaction evaluation of emergency call IVRs has been minimally researched by the scientific community.

Author Contributions: Conceptualization, B.-C.M., I.-D.F. and F.P.; methodology, R.-D.U., C.N. and I.B.; software, B.-C.M., R.-D.U. and T.-C.B.; validation, I.-D.F., C.N. and M.D.; formal analysis, M.D. and S.-A.T.; investigation, M.D. and S.-A.T.; resources, M.D. and S.-A.T.; data curation, R.-D.U. and I.-D.F.; writing—original draft preparation, B.-C.M., C.N. and S.-A.T.; writing—review and editing, I.-D.F. and C.N.; visualization, B.-C.M. and R.-D.U.; supervision, T.-C.B., I.B. and F.P.; project administration, T.-C.B., I.B. and F.Pop.; funding acquisition, T.-C.B., I.B. and F.P. All authors have read and agreed to the published version of the manuscript.

Funding: The research presented in this paper was supported by the project ODIN112 (PN-III-P2-2.1-SOL-2021-2-0223) and by the University Politehnica of Bucharest through the PubArt program.

This work was supported in part by OPTIM Research (POCU grant no. 62461/03.06.2022, cod SMIS: 153735).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We would like to thank the reviewers for their time and expertise, constructive comments, and valuable insight.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial intelligence
AIML	Artificial Intelligence Markup Language
CVA	Cerebral vascular accident
DFTM	Dual-tone multi-frequency signaling
FEM	Finite Element Method
GUI	Graphical user interface
HCI	Human–computer interaction
IVR	Interactive voice response
IoT	Internet of Things
NLP	Natural language processor
NLU	Natural language understanding
PoC	Proof of concept
PSAP	Public-service answer point
QoS	Quality of Service
RTD	Round-trip delay
RTCP	Real-time control protocol
SDK	Software development kit
SIP	Session initiation protocol
SLA	Service level agreement

References

1. Dahiya, M. A tool of conversation: Chatbot. *Int. J. Comput. Sci. Eng.* **2017**, *5*, 158–161.
2. Luo, B.; Lau, R.Y.; Li, C.; Si, Y.W. A critical review of state-of-the-art chatbot designs and applications. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2022**, *12*, e1434. [CrossRef]
3. Cheong, J.; Tucker, J.A. Utility of interactive voice response self-monitoring in stabilizing initial change during natural recovery attempts among persons with alcohol use disorder. *J. Subst. Abus. Treat.* **2022**, *140*, 108831. [CrossRef] [PubMed]
4. Tsoli, S.; Sutton, S.; Kassavou, A. Interactive voice response interventions targeting behaviour change: A systematic literature review with meta-analysis and meta-regression. *BMJ Open* **2018**, *8*, e018974. [CrossRef] [PubMed]
5. Tan, R.Y.H.; Lee, C.S.; Pichika, M.R.; Cheng, S.F.; Lam, K.Y. PH Responsive Polyurethane for the Advancement of Biomedical and Drug Delivery. *Polymers* **2022**, *14*, 1672. [CrossRef]
6. Adamopoulou, E.; Moussiades, L. An overview of chatbot technology. In Proceedings of the IFIP International Conference on Artificial Intelligence Applications and Innovations, Halkidiki, Greece, 5–7 June 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 373–383.
7. Madsen, J.L.; Lauridsen, K.G.; Løfgren, B. In-hospital cardiac arrest call procedures and delays of the cardiac arrest team: A nationwide study. *Resusc. Plus* **2021**, *5*, 100087. [CrossRef]
8. French, R.M. The Turing Test: The first 50 years. *Trends Cogn. Sci.* **2000**, *4*, 115–122. [CrossRef]
9. Weizenbaum, J. ELIZA—a computer program for the study of natural language communication between man and machine. *Commun. ACM* **1966**, *9*, 36–45. [CrossRef]
10. Marzavan, S.; Nastasescu, V. Displacement calculus of the functionally graded plates by finite element method. *Alex. Eng. J.* **2022**, *61*, 12075–12090. [CrossRef]
11. Marzavan, S. EFG method in numerical analysis of foam materials. *Mech. Time-Depend. Mater.* **2022**, *26*, 409–429. [CrossRef]
12. Colby, K.M.; Weber, S.; Hilf, F.D. Artificial paranoia. *Artif. Intell.* **1971**, *2*, 1–25. [CrossRef]
13. Wallace, R.S. The anatomy of ALICE. In *Parsing the Turing Test*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 181–210.

14. Ramesh, K.; Ravishankaran, S.; Joshi, A.; Chandrasekaran, K. A survey of design techniques for conversational agents. In Proceedings of the International Conference on Information, Communication and Computing Technology, Singapore, 27–29 December 2017; Springer: Berlin/Heidelberg, Germany, 2017; pp. 336–350.
15. Marietto, M.d.G.B.; de Aguiar, R.V.; Barbosa, G.d.O.; Botelho, W.T.; Pimentel, E.; França, R.d.S.; da Silva, V.L. Artificial intelligence markup language: A brief tutorial. *arXiv* **2013**, arXiv:1307.3091.
16. Nuruzzaman, M.; Hussain, O.K. A survey on chatbot implementation in customer service industry through deep neural networks. In Proceedings of the 2018 IEEE 15th International Conference on e-Business Engineering (ICEBE), Xi'an, China, 12–14 October 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 54–61.
17. Jung, S. Semantic vector learning for natural language understanding. *Comput. Speech Lang.* **2019**, *56*, 130–145. [CrossRef]
18. Bocklisch, T.; Faulkner, J.; Pawlowski, N.; Nichol, A. Rasa: Open source language understanding and dialogue management. *arXiv* **2017**, arXiv:1712.05181.
19. Al-Sinani, A.H.; Al-Saidi, B.S. A Survey of Chatbot creation tools for non-coder. *J. Stud. Res.* **2019**. [CrossRef]
20. Nenciu, B.; Corlatescu, D.G.; Dascalu, M. RASA Conversational Agent in Romanian for Predefined Microworlds. In Proceedings of the International Conference on Human-Computer Interaction (RoCHI2020), Online, 22–23 October 2020; MatrixRom: Bucuresti, Romania, 2020.
21. Bunk, T.; Varshneya, D.; Vlasov, V.; Nichol, A. Diet: Lightweight language understanding for dialogue systems. *arXiv* **2020**, arXiv:2004.09936.
22. Boroghina, G.; Corlatescu, D.G.; Dascalu, M. Conversational Agent in Romanian for Storing User Information in a Knowledge Graph. In Proceedings of the International Conference on Human-Computer Interaction (RoCHI2020), Online, 22–23 October 2020; MatrixRom: Bucuresti, Romania, 2020.
23. Messina, A.; Pribadi, H.; Stichbury, J.; Bucci, M.; Klarman, S.; Urso, A. BioGrakn: A knowledge graph-based semantic database for biomedical sciences. In Proceedings of the Conference on Complex, Intelligent, and Software Intensive Systems, Torino, Italy, 10–12 July 2017; Springer: Berlin/Heidelberg, Germany, 2017; pp. 299–309.
24. Belda-Medina, J.; Calvo-Ferrer, J.R. Using Chatbots as AI Conversational Partners in Language Learning. *Appl. Sci.* **2022**, *12*, 8427. [CrossRef]
25. Mageira, K.; Pittou, D.; Papasalouros, A.; Kotis, K.; Zangogianni, P.; Daradoumis, A. Educational AI Chatbots for Content and Language Integrated Learning. *Appl. Sci.* **2022**, *12*, 3239. [CrossRef]
26. Zygadło, A.; Kozłowski, M.; Janicki, A. Text-Based emotion recognition in English and Polish for therapeutic chatbot. *Appl. Sci.* **2021**, *11*, 10146. [CrossRef]
27. Biswas, M. Microsoft Bot Framework. In *Beginning AI Bot Frameworks*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 25–66.
28. Williams, J.D.; Kamal, E.; Ashour, M.; Amr, H.; Miller, J.; Zweig, G. Fast and easy language understanding for dialog systems with Microsoft Language Understanding Intelligent Service (LUIS). In Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, Prague, Czech Republic, 2–4 September 2015; pp. 159–161.
29. Tri, T.C.M. Creating Smart, Human-Like Chatbot for Businesses Using BotPress Platform. PhD Thesis, German University, Saitama, Germany, 2020. (In Vietnamese)
30. Haldén, F.J.; Yao Håkansson, J. *A Comparison between Chatbots for Handling Administrative Healthcare Tasks*; KTH: Stockholm, Sweden, 2020.
31. Theosaksomo, D.; Widyantoro, D.H. Conversational Recommender System Chatbot Based on Functional Requirement. In Proceedings of the 2019 IEEE 13th International Conference on Telecommunication Systems, Services, and Applications (TSSA), Bali, Indonesia, 3–4 October 2019; pp. 154–159.
32. Malas, D.; Morton, A. *Basic Telephony Sip End-to-End Performance Metrics*; Technical Report; AT&T Labs: Atlanta, GA, USA, 2011.
33. Klüwer, T. From chatbots to dialog systems. In *Conversational Agents and Natural Language Interaction: Techniques and Effective Practices*; IGI Global: Hershey, PA, USA, 2011; pp. 1–22.
34. Eeuwen, M.v. Mobile Conversational Commerce: Messenger Chatbots as the Next Interface between Businesses and Consumers. Master's Thesis, University of Twente, Enschede, The Netherland, 2017.
35. Ramos, R. Screw the Turing Test-Chatbots don't need to act human. *Ventur. Retrieved March* **2017**, *13*, 2017.
36. Bostrom, N.; Yudkowsky, E. The ethics of artificial intelligence. In *Artificial Intelligence Safety and Security*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2018; pp. 57–69.
37. Morrissey, K.; Kirakowski, J. 'Realness' in chatbots: Establishing quantifiable criteria. In Proceedings of the International conference on human-computer interaction, Cape Town, South Africa, 2–6 September 2013; Springer: Berlin/Heidelberg, Germany, 2013; pp. 87–96.
38. Kudashkina, K.; Pilarski, P.M.; Sutton, R.S. Document-editing assistants and model-based reinforcement learning as a path to conversational AI. *arXiv* **2020**, arXiv:2008.12095.
39. Minessale, A.; Schreiber, D. *FreeSWITCH Cookbook*; Packt Publishing Ltd.: Birmingham, UK, 2012.
40. Zadeh, L.A. Fuzzy logic. *Computer* **1988**, *21*, 83–93. [CrossRef]
41. Ierusalimschy, R. (Ed.) *Programming in Lua*; Lua.org: Rio de Janeiro, Brazil, 2006.
42. Azrour, M.; Farhaoui, Y.; Guezzaz, A. Experimental validation of new SIP authentication protocol. In Proceedings of the International Conference on Big Data and Networks Technologies, Harbin, China, 19–20 December 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 1–11.