

Go-Omni: a lightweight infrastructure for an omnichannel service

Armina Ibrić Imamović¹, Edin Pjanić²

¹*Bicom doo Tuzla, Bosnia and Herzegovina*

²*Faculty of Electrical Engineering, University of Tuzla, Bosnia and Herzegovina*

Email: armina.ibric@gmail.com

Abstract. One of the latest trends nowadays in online shopping, retail and contact centers is omnichannel. It is a strategy that provides a seamless and integrated customer experience across all available communication channels. It is used in contact centers to improve their customer service by increasing their efficiency and by optimising their system performance. The paper presents the infrastructure of an omnichannel service enabling a simple integration of messaging channels. The service supports different login authenticators, to ensure its usability for various purposes. The current implementation of the login authenticator uses Asterisk *PBX*, which enables the implementation of a contact center system. The omnichannel service over a *TCP* connection provides *API* for exchanging messages between identified service users and clients over messaging channels.

Keywords: Omnichannel, Multichannel, Contact Center, Customer Journey, Customer Experience

Go-Omni: lahka infrastruktura za večkanalno storitev

Eden najnovjših trendov današnjega časa pri spletnem nakupovanju, maloprodaji in kontaktnih centrih je večkanalni način. To je strategija, ki zagotavlja brezhibno in integrirano uporabniško izkušnjo v vseh razpoložljivih komunikacijskih kanalih. Uporablja se v kontaktnih centrih za izboljšanje njihovih storitev za stranke s povečanjem njihove učinkovitosti in optimizacijo delovanja njihovega sistema. Prispevek predstavlja infrastrukturo večkanalne storitve, ki omogoča preprosto integracijo sporočilnih kanalov. Storitve podpira različne avtentikatorje za prijavo, da se zagotovi njena uporabnost za različne namene. Trenutna izvedba prijavnega avtentifikatorja uporablja Asterisk *PBX*, ki omogoča implementacijo sistema kontaktnega centra. Večkanalna storitev prek povezave *TCP* zagotavlja *API* za izmenjavo sporočil med identificiranimi uporabniki storitve in odjemalci prek kanalov za sporočanje.

1 INTRODUCTION

Social media are the most popular communication channel worldwide. Social media have gone beyond being just an online communication platform. They have become a channel that users prefer to other channels such as calls, email, SMS, etc. They are also a popular way of communicating with service providers. Customers can start a conversation with a service provider on one social media and quickly switch to another in order to get a most precise, accurate and customized service possible [1].

The Internet, smart devices and social media are the main drivers that bring a new era in the customer service and the most significant changes in contact centers [2].

Traditionally, call centers provide services on only one communication channel, i.e., a phone call. The evolution of the market and digital technologies has led to the need to provide customers with additional communication channels. Many companies offer different channels through which customers can contact them, such as phone, SMS, email, social networks and web chat. Such companies are considered to be multichannel companies. When all the different channels of their communication system become a unique experience for customers and agents, then it is an omnichannel approach.

Nowadays, an omnichannel approach is necessary for the company contact centers because it improves their customer service. Customers expect a consistent, uniform and integrated customer service [3]. They want to move seamlessly between channels during the interaction without having to repeat themselves [4]. Customers expect the customer service to be consistent across all the platforms, personalized to their demands and requiring a little effort from their side. An omnichannel approach allows customers to interact with the service provider in the way they want and through the channel they prefer at a certain moment in time. To provide a seamless customer experience, companies must

keep as many channels as possible open for their customers. One of the main challenges when applying the omnichannel approach in contact centers is implementing a seamless and integrated customer experience [5].

The paper presents an omnichannel service Go-Omni that facilitates a seamless integration of messaging communication channels in a simple and user-friendly manner. It is originally designed for contact center systems. The service supports various login authenticators, enabling it to be used for diverse purposes.

The paper is organized as follows. The first chapter delves into the significance of the omnichannel approach and provides an overview of previous research relevant to the subject. The second chapter contains a concise description of the omnichannel approach. It traces the evolution from traditional call center systems to modern omnichannel contact center systems. It explains the fundamentals of Asterisk PBX as the communication platform for implementing the contact center and provides an insight into the commonly used messaging channels and their integration procedures. The third chapter presents the infrastructure for the implemented omnichannel service and the measured system performance. The paper concludes with a summary.

1.1 Related work

Until recently, the main area where the omnichannel approach was relevant was retailing, marketing and e-commerce [6]. Many papers can be found in the context of omnichannel retailing. C. Lazaris and A. Vrechopoulos [6] explore evolving retail practices and consumer behaviour driven by advanced technologies. They emphasize the transition from the multichannel to the omnichannel retailing, advocating for a comprehensive multidisciplinary research to understand the impact of information technologies on modern business and customer behaviour. [7] identifies the factors affecting the omnichannel customer behaviour, particularly their acceptance of and intention to use new technologies during shopping. G. Cook [8] discusses the nature of the omnichannel customers and the transformations occurring in physical retail spaces. He also addresses the need for new technologies to support the store staff and introduces novel store performance metrics. [4] provides a valuable contribution to understanding the role of the information technology in retailing. It delves into crucial aspects of omnichannel retailing, including channel integration and the impact of

the mobile technology and addresses diverse customer requirements, the significance of the social media, and the evolution of physical store functions. [9] delves into the study of omnichannel customers and their behaviour patterns related to a simultaneous use of different shopping channels.

There is not much literature available on the omnichannel approach to be used in contact centers. There are papers that focus on problems of contact centers that need to be solved in order to implement an omnichannel approach. L. Bocklund [10] focuses on the potential challenges during the transition from a multichannel to an omnichannel contact center, as well as the advantages it brings. [11] identifies four critical user identification-related issues to be addressed by the contact centers to improve their performance and achieve a greater efficiency. [1] analyses the most widely used social media and the types of the personal data they gather. Moreover, the paper explains the fundamental principles of the GDPR (*General Data Protection Regulation*) that must be adhered to when collecting the data from the social media. R. Picek et al [2] discuss the significance of implementing an omnichannel approach in contact centers and highlight the opportunities and challenges that such an approach presents.

2 OMNICHANNEL APPROACH

"*OMNIS*" is a Latin word that means "all" or "universal" thus the "omnichannel" can be interpreted as "all channels" [6]. The omnichannel approach represents a strategy in which customers can communicate with a service provider through all available communication channels [2]. To become an omnichannel company, it is not enough to simply possess multiple communication channels. It is necessary for all channels to be integrated. Instead of multiple channels, the omnichannel is actually a single channel with multiple touchpoints that provides a seamless and consistent customer experience [4]. All that customers are really concerned about are not the channels but finding an answer to their current needs or desires, through a convenient and valuable experience that optimizes their time and money. To deliver a seamless customer experience, it is essential to integrate data from all available communications channels in order to create and analyze customer profiles.

2.1 Omnichannel contact center vs. call center

A call center is a customer service operation that provides a sales and service support through inbound and/or outbound phone calls. It is staffed with specially trained customer service agents who handle customer inquiries but may also provide additional services such as telemarketing, debt collection, and other functions related to a customer support. Call centers only provide a phone support, which is the main difference compared to the contact centers.

To meet the customers expectations for a fast, personalized, and timely response across their preferred channels, companies often have specialized departments that manage both the incoming and outgoing customers interactions. These departments cover various communication channels, such as phone calls, text, email, live chat and social media, and are called contact centers. When all the channels are integrated, it is known as an omnichannel approach. While an omnichannel contact center may not be functional on all communication channels, it ensures that the channels it operates on are interconnected and integrated, providing customers with a seamless experience.

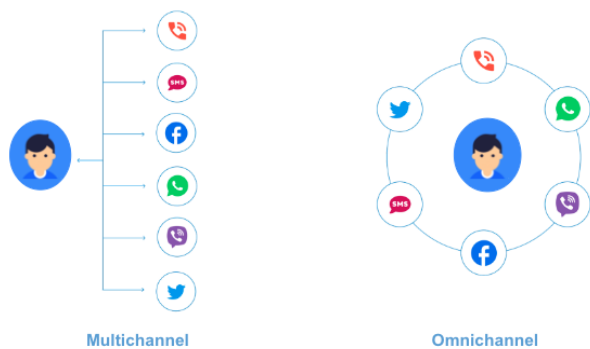


Figure 1. Multichannel vs. omnichannel contact center.

Implementing an omnichannel approach offers advantages to both the customers and agents [2]. Agents are provided with a unified *GUI* interface that accommodates all communication channels, considerably easing the receipt and processing of diverse requests from varying channels. They can review a complete customer journey encompassing all previous interactions across every channel, facilitating an effective problem resolution and reducing the time spent addressing inquiries. These advantages collectively boost the efficiency and performance of the contact center system.

To establish an omnichannel contact center, the major components include a central communication management platform, such as Asterisk, messaging channels for engaging with customers, and customers identification to improve personalized services.

2.2 Asterisk PBX

Asterisk is a free, open-source *VOIP PBX* software used for implementing communication applications [12]. Asterisk is widely used in contact center systems and is considered to be the most successful *PBX (Private Branch Exchange)* in the world. *PBX* is a private telephone network used within a company [13]. Asterisk is used by small and large companies, call centers, carriers and government organizations worldwide for building a powerful, scalable business telephone system (*PBX*) or as an upgrade to enhance the existing systems. One of the following interfaces can be used to communicate with Asterisk:

- **AGI (Asterisk Gateway Interface)** allows external programs to manage Asterisk resources.
- **AMI (Asterisk Manager Interface)** is used for system monitoring and management.
- **ARI (Asterisk Rest Interface)** enables the implementation of a communication application using primitive Asterisk objects such as channels, bridges, media, etc.

2.3 Messaging channels

The evolution of the social media and communication channels has changed the way customers expect to contact a company. They want to be able to reach a company through their favourite messaging applications. To meet their needs, companies are rapidly working to provide a support on the communication channels that their customers prefer. While it is important for companies to keep as many communication channels as possible open, they prioritize the channels that are most important to their customers at any given time. The social media is currently the most commonly used messaging channel worldwide.

Integrating a messaging channel is a multi-stage process. First, a user account, page, or bot must be configured to enable the company to be available to its customers. The specific steps for configuring the messaging channel vary for each platform. For instance, integrating with *WhatsApp* requires the use of a communication platform that supports it, such as *Twilio*, *Telnyx*, or *360Dialog*. The user account on the selected

platform is then used to communicate with the *WhatsApp* application via its *API*. *Viber* integration requires creating a *Viber Bot* account (free version), or a *Viber Business* account (paid version) to enable a messaging exchange.

Once the configuration is complete, the next step is the same for all messaging channels: setting up a webhook. It is an *HTTP*-based callback function that allows a lightweight, event-driven communication between two application programming interfaces (*APIs*). It facilitates a communication with the messaging channel by enabling the sending and receiving of messages.

2.4 Customer identity

In order to provide the best possible experience for customers and agents, and to increase the performance of the contact centers, it is necessary to implement a conversation history for the customer across all available channels. The end-to-end experience and interactions a customer has with a company across multiple touchpoints and channels is called the customer journey.

To identify and implement the customer journey, it is necessary to have information that is unique to the customer for a specific communication channel. Table 1 shows how the customer identity is determined for some communication channels.

Table 1. Unique customer identifiers for different channels.

Communication channel	Customer identity
Phone Call	Phone number
SMS	Phone number
WhatsApp	Phone number
Viber	1. Name/Viber ID (if using free Viber Bot approach) 2. Name/Phone number (if using paid Viber Business account)
Facebook	Name/Facebook User ID
Twitter	Name/Twitter User ID

Each channel has a unique identifier for each customer. The customer journey for each channel is easy to implement. Different communication channels have different unique identifiers for the customer, such as a number, name, customer ID, etc. [2]. To implement the customer journey for all channels, it is necessary to integrate the received data from all channels and create customer profiles.

3 SYSTEM DESCRIPTION

The main goal of the Go-Omni omnichannel service is to enable an easy integration of each messaging channel. This service currently supports two messaging channels, WhatsApp and Viber. However, it can be easily adapted to include any other preferred channels. A context diagram illustrating the omnichannel service is presented in Figure 2.

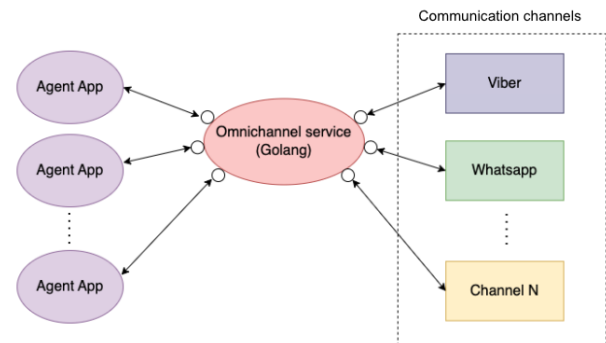


Figure 2. Context diagram for the Go-Omni omnichannel service .

The omnichannel service consists of the following components:

- HTTP server - a message exchange and communication with various messaging channels.
- Channel Interface - a seamless addition and integration of new messaging channels.
- TCP server - communication with authenticated system users.
- LoginAuthenticator Interface - implementation of custom login authentication methods ensuring adaptability for a specific use cases.
- DBConnector - a pivotal component for database operations such as message storage and customer information management.

An in-depth explanation of the above components is given below.

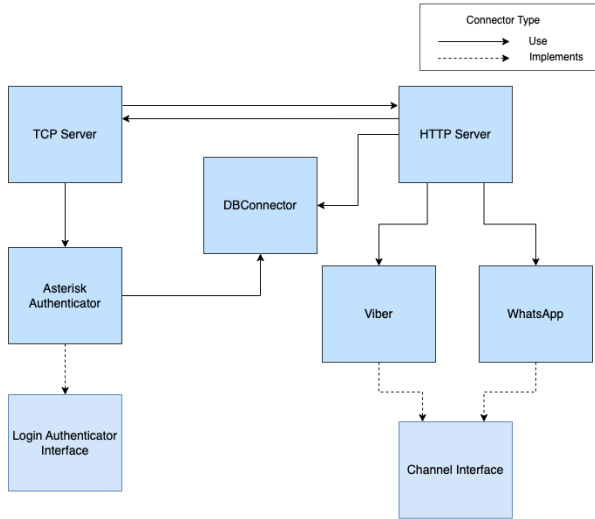


Figure 3. Components of the omnichannel service.

The omnichannel service is developed using the Golang programming language. It is a popular choice for building high-performance services and applications due to its efficiency, speed, and robust concurrency support. Its lightweight goroutines and channels facilitate an efficient management of concurrent requests. Its standard library provides a comprehensive HTTP package for a simplified development. The HTTP server efficiently manages multiple concurrent requests by employing goroutines, with each incoming request being processed in its dedicated goroutine, ensuring an effective multitasking with a minimal resource overhead.

3.1 The message processing details

In the context of the omnichannel service, the HTTP server has a dual role. It handles the communication with messaging channels and processes incoming messages from specific channels. Figure 4 shows its state diagram. The Golang language built in the concurrency capabilities enables the server to efficiently handle multiple messages simultaneously. To ensure the service to operate efficiently, is essential to initialize both the database and the supported messaging channels before starting the HTTP server.

The flowchart in Figure 5 shows the message processing procedure for the incoming messages received from a messaging channel.

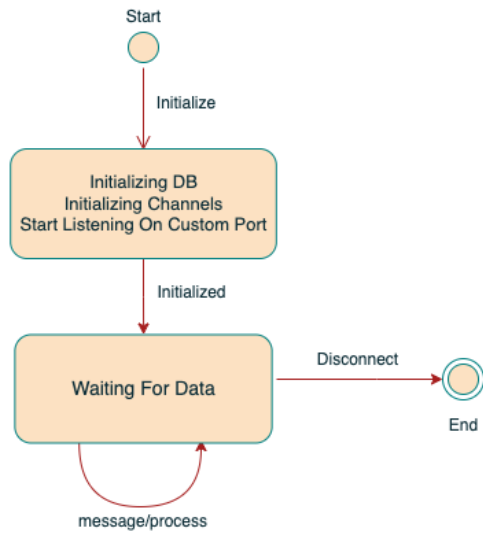


Figure 4. State diagram for the HTTP server.

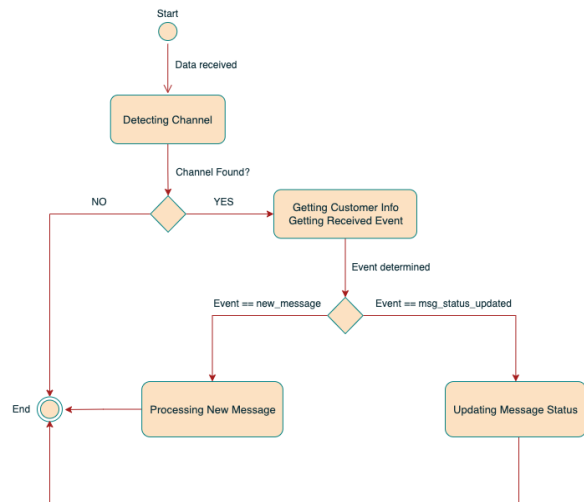


Figure 5. Flow diagram for processing messages.

When a new message is received from a channel, the first step is to confirm if the service supports the integration with that channel. If it does, the next step is to determine the identity of the customer using the methods described earlier. After this, the message is processed based on the event received. This involves adding a new message within a conversation or updating the status (sent, delivered, seen) of an existing message.

3.2 Integration of a new channel

To integrate a new messaging channel for communication to the omnichannel service, it is necessary to create a new class that implements all the methods of the Channel interface (see Figure 6).

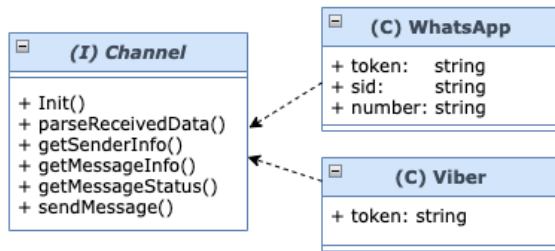


Figure 6. *Channel* interface and integrated channels.

3.3 Customer journey

The omnichannel service uses the *MySQL* database to store the information about new customers, new communication channels of the existing customers, conversations and messages. The structure of the database is shown in Figure 7. The *customer* table stores the unique identifier and name for each customer. A customer can communicate with the company multiple times using different channels. The *customer_contacts* table stores the information about all channels through which a customer communicates with the company. The *conversations* table stores conversations from all customers, while the *messages* table stores messages from all conversations.

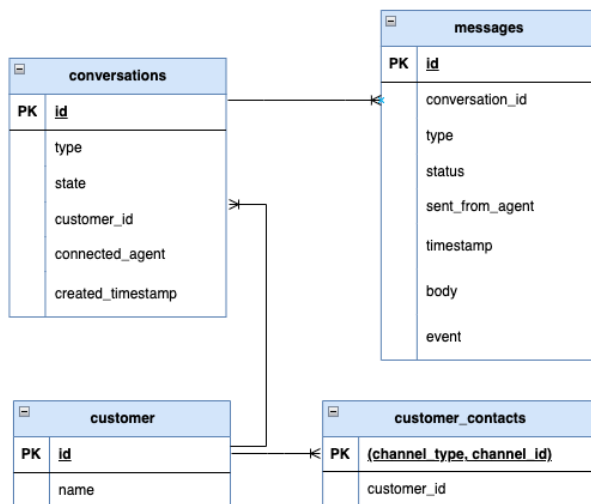


Figure 7. Database structure.

By having an access to the customer history through the application, agents can quickly understand the context of the conversation and provide a more personalized and effective customer service experience. To implement a customer journey for all communication channels, it is necessary to have a unique information of the customer across the channels. While different messaging channels may have different unique identifiers for customers, such as phone numbers for WhatsApp or ViberID for Viber, a common identifier

must be introduced to connect conversations across channels and create a customer conversation history. In this solution, the customer's name is used as a common identifier. When a customer initiates a conversation on some messaging channel for the first time, it is important to store relevant information about the customer and the channel used in the conversation in the database. This data is used to identify the customer and enables the retrieval of their conversation history in future interactions. To avoid duplicating the customer information in the database, it is verified whether the customer already exists in the database before saving their information. The customer search within the database is performed based on the customer's name. If the customer already exists, only new contact information needs to be added to the database. This ensures that the company has a complete record of all customer interactions while maintaining the data integrity. The current implementation of the omnichannel service is based on the approach of storing the customer information based on their name. However, this approach can be unreliable as customers often use only their first name or a nickname on social media accounts. To improve this, the following methods can be employed:

- Involve agents in the process of collecting the customer information. Agents can request relevant information during the customer communication and update the customer data as needed.
- Implement automated bot messages that request the necessary information from the customer at the start of a new conversation.

It is important to note that in implementing the customer journey, the customer plays an important role. If a customer does not want to share their personal information, it may not be possible to fully implement the customer journey for all channels and maintain a 100% reliability. Moreover, the protection of the customer's personal data should be a top priority, and the purpose of collecting the information should be clearly explained to the customers, with their consent obtained before storing their data [1].

3.4 Authentication

The omnichannel service provides an *API* via a *TCP* connection used for a message exchange between authenticated service users and customers from supported messaging channels. Since this service was originally intended for the use in contact center systems, the current login authenticator is based on Asterisk. It

uses the *AMI* module to communicate with Asterisk and implements the user login/logout. In this way, each user who logs in to the omnichannel service will be logged in as an agent on the Asterisk system and will be able to receive phone calls. Of course, in order to use this service for contact centers, it is necessary to configure Asterisk and implement a *GUI* application that supports a *SIP* softphone for receiving calls, as well as an interface for the sending/receiving messages.

To implement a new login authenticator, it is necessary to create a new class that implements all the methods of the *LoginAuthenticator* interface (refer to Figure 8). Afterwards, it should be configured within the "*InitializeLoginAuthenticator*" function.

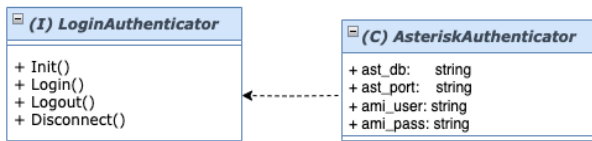


Figure 8. *LoginAuthenticator* interface.

3.5 System performance

The chart in Figure 9 shows the systems processing speed for handling new messages, which is measured to assess the system performance. The measurements are conducted on a macOS system powered by a 2.6 GHz 6-Core Intel Core i7 processor and performed with a certain number of burst messages. To send a certain number of burst messages, a test application is implemented that generates messages in a format expected by the service. The total execution time for processing these messages is calculated as follows:

- *StartTime* - the moment when the initial message of a certain number of burst messages is received.
- *EndTime* - the moment when the last message within a certain number of burst messages is successfully processed.
- *ExecutionTime* - the overall time taken to process a certain number of burst messages. It is determined by calculating the time elapsed between *StartTime* and *EndTime*.

The results illustrate the average execution time for a certain number of burst messages based on ten separate measurements.

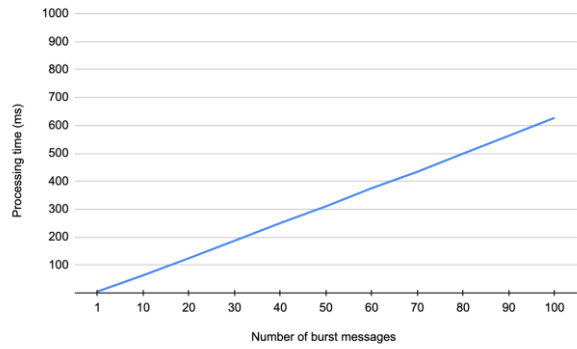


Figure 9. Function processing time graph with a database.

The processing time for a single message, either the only incoming message or part of a burst of 100 incoming messages, is roughly the same. This might be counterintuitive, since the messages are processed simultaneously in separate goroutines. However, two key factors affect the outcome. The first is the impact of the concurrency overhead when processing multiple messages concurrently. The overhead involves managing and switching between multiple goroutines, which can slightly increase the processing time. The second factor is the presence of a bottleneck of the processing function. Since the processing function interacts with the database, a concurrent access by multiple goroutines can lead to database locking or contention. In Golang, most database drivers operate synchronously by default. This means that when multiple goroutines attempt to read or write to the database concurrently, they tend to execute in a serial manner. This serialization is enforced through connection pooling in most database drivers to maintain the data consistency.

Due to the presence of the bottleneck caused by the database interactions, measurements are also performed without using the database. The results of the measurements are quite impressive. 100 messages can be processed in just 9 ms when the database is turned off. The measurements are shown in Figure 10.

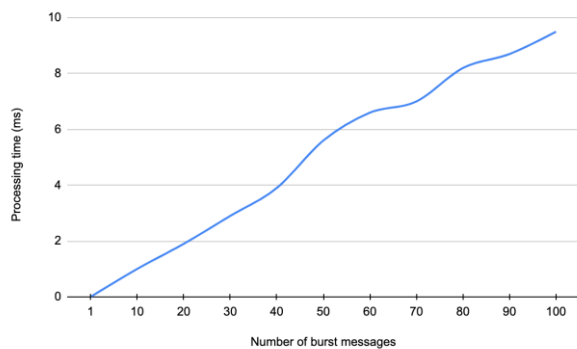


Figure 10. Function processing time graph without the database.

The need for a database in an omnichannel service depends on its intended use. If a service is primarily for a real-time message exchange and requires no conversation history storage, a database may not be necessary. However, in scenarios where a persistent data storage is required, various solutions can be considered. Certain Golang database libraries or database management systems provide asynchronous capabilities or non-blocking database drivers. In such cases, configuring database interactions to be asynchronous enables multiple goroutines to initiate the database operations concurrently, without waiting for a prior operations completion. Such configuration has a potential to improve the system concurrency and performance.

4 CONCLUSION

The omnichannel approach has a significant role in modern business and is important for the success of companies. The omnichannel service presented in this paper is a solution that enables an easy integration of any preferred messaging channel. It supports multiple authentication methods and is adaptable to various purposes. Originally developed for an integration within contact center systems using an Asterisk-based authentication, this service can be seamlessly adapted for a wide range of applications. For instance, it is well-suited for building applications that enable a real-time message exchange across diverse communication channels. The service is designed for a speedy message processing, capable of processing 100 simultaneous messages in less than one second. In line with the omnichannel approach, the service implements the customer journey across all channels, making it possible to achieve the primary goal of the omnichannel approach. The implementation of the omnichannel service is publicly accessible at <https://github.com/ArmUSER/Go-Omni>.

It can serve as a guide for implementing similar solutions or upgrading an existing service.

The paper contributes to a deeper understanding and practical application of the omnichannel approach in customer communication and paves the way for further research and improvement in this important aspect of modern business.

REFERENCES

- [1] D. Peras, et al. "Influence of GDPR on social networks used by omnichannel contact center." *MIPRO*, 2018.
- [2] R. Picek, et al. "Opportunities and Challenges of Applying Omnichannel Approach to Contact Center." *IEEE International Conference on Information Management*, 2018.
- [3] P. Rebaque-Rivas, and E. Gil-Rodriguez. "Adopting an Omnichannel Approach to Improve User Experience in Online Enrolment at an E-learning University." *HCI International 2019 - Posters*.
- [4] W. Piotrowicz and R. Cuthbertson. "Introduction to the Special Issue Information Technology in Retail: Toward Omnichannel Retailing." *Int. J. Electron. Commer.*, 2014.
- [5] S. Peltola, et al. "Key factors in developing omnichannel customer experience with finnish retailers." *HCIB*, vol. 2015.
- [6] C. Lazaris, and A. Vrechopoulos. "From Multichannel to 'Omnichannel' Retailing: Review of the Literature and Calls for Research." *Unpublished*, 2014.
- [7] E. Juaneda-Ayensa, et al. "Omnichannel Customer Behavior: Key Drivers of Technology Acceptance and Use and Their Effects on Purchase Intention." *Front Psychol*, 2016.
- [8] G. Cook. "Customer experience in the omnichannel world and the challenges and opportunities this presents." *Macmillan Publ. LTD*, 2014.
- [9] C. Lazaris, et al. "Exploring the 'Omnichannel' Shopper Behaviour." *Unpublished*, 2014.
- [10] L. Bocklund. "The Multichannel Contact Center Becomes 'Ominchannel.'" *Pipeline Artic*, 2015.
- [11] M. Saberi, et al. "Challenges in Efficient Customer Recognition in Contact Centre: State-of-theArt Survey by Focusing on Big Data Techniques Applicability." 2016.
- [12] Leif Madsen, et al. *Asterisk: The Definitive Guide, 5th Edition*. O'Reilly Media, Inc., 2019.
- [13] A. Imran, et al. "Asterisk VoIP Private Branch Exchange." *IEEE 2009 International Multimedia, IMPACT 2009*, 2009.

Armina Ibrić Imamović received her B.Sc. degree in Electrical Engineering from the Faculty of Electrical Engineering, University of Tuzla, Bosnia and Herzegovina, in 2016. Currently, she works with the Bicom Systems d.o.o. as a software engineer. Her research involves omnichannel strategy, contact center systems and software engineering.

Edin Pjanić received his Ph.D. degree in 2012 from the Faculty of Electrical Engineering, University of Tuzla, Bosnia and Herzegovina, where he is currently an Associate Professor. His research interests are in software engineering and development, domain specific languages, web, education in computer science.