Towards the Design of an Intelligent Location-Aware Architecture for Mobile Computing Environments

Santipong Thaiprayoon
Chair of Communication Networks
FernUniversität in Hagen
Hagen, Germany
santipong.thaiprayoon@fernuni-hagen.de

Herwig Unger
Chair of Communication Networks
FernUniversität in Hagen
Hagen, Germany
herwig.unger@fernuni-hagen.de

Abstract—**The pervasiveness of mobile computing has become ubiquitous in daily life, particularly for people who visit various locations in the physical world through their mobile devices. Similarly, providing valuable services, actions, and information tailored to a specific area or environment can make user interactions more seamless, personalized, and convenient. The article proposes an intelligent location-aware architecture for mobile computing environments that interacts with individual mobile users by automatically providing local services, helpful information, and personalized recommendations based on their current location and user context. Experimental results show that the proposed architecture is intelligent, privacy-preserving, scalable, reliable, and can be efficiently deployed in various real-world environments.**

***Index Terms*—Location-aware systems; Context awareness; Personalization; Mobile computing.**

I. Introduction

The article is an extension of research work that was originally proposed in the Fifteenth International Conference on Advances in Future Internet (AFIN 2023) [1], which provided the design of an intelligent location-aware architecture for mobile computing environments.

The evolution of mobile computing has dramatically changed how individuals interact with their surroundings and access information using mobile devices or wearable devices, such as smartphones, smartwatches, and tablets. Mobile computing holds the vision for adaptation in an environment where users can seamlessly access information and services based on their location and preferences [2], [3]. This vision has led to the development of platforms for a myriad of applications, such as Location-Based Services (LBSs), Augmented Reality (AR), and context-aware applications. Most of these applications utilize wireless communication technologies, like Global Positioning System (GPS), Wireless Fidelity (Wi-Fi), Radio-Frequency Identification (RFID), and Bluetooth Low Energy (BLE). The convergence of these wireless communication technologies also opens new possibilities for Location-Aware Systems (LASs) [4], [5], aiming to determine the location of users in real-time and provide information and services to users.

LASs [6]–[8] are applications that have been widely used in various domains, including tourism, transportation, healthcare, education, entertainment, and social networking. They can sense and react to the physical location of users and devices and provide them with relevant and timely information or services. For example, a location-aware system in the tourism industry can provide users with useful recommendations for nearby attractions, restaurants, and accommodations based on their current location. This not only enhances the user experience but also enables users to discover new places and events that are relevant to their interests and preferences.

Despite the advancements in LASs, there are still significant challenges in effectively deploying these systems [9]–[11], particularly in specific environments, such as shopping malls, museums, or university campuses. As a result, these environments demand a tailored approach that considers unique opportunities and requirements based on location data and individual preferences. Likewise, most existing LBS platforms are designed for general-purpose scenarios and do not account for the specific characteristics and needs of different environments and users. Furthermore, these systems lack more personalization and contextual awareness, leading to a failure to meet the unique needs of individual users. Additionally, there is a need to address challenges in accuracy, security concerns, privacy protection, scalability, and user experience related to diverse user preferences and behaviors.

To address these challenges, this article proposes a novel conceptual architecture for intelligent location-aware systems optimized for specific environments. The proposed architecture aims to enable individual mobile users to interact with their specific environments through their mobile devices, enhancing the overall user experience in different contexts through personalized interaction and services. This opens an opportunity to develop a system that not only understands where a user is but also who the user is in terms of user context and needs. The proposed architecture is designed to recognize and adapt to the unique characteristics of specific environments, offering a suite of localized services, useful information, and personalized recommendations by utilizing the current context of individual mobile users, including location data and user profiles. The integration of several advancing fields, such as mobile computing, Artificial Intelligence (AI), Machine Learning (ML), Natural Language Processing (NLP), and data analytics, leverages a deep understanding of the specific context and dynamics of these environments, proving more effective than general-purpose LBS platforms. This combination helps deliver more intelligent services that are seamlessly integrated into the daily activities of mobile users.

The main contributions of this article are summarized as follows:

- The proposed architecture aims to enhance the field of location-aware systems by addressing current limitations in terms of accuracy, privacy, security, scalability, and personalization.
- This article offers valuable insights into designing and implementing a novel conceptual architecture for a location-aware

system, particularly tailored to specific environments based on the current location and context of individual users.
- Extensive experiments are conducted to validate the robustness and effectiveness of the proposed architecture.

The remainder of this article is organized as follows: Section II offers a comprehensive review of related literature, highlighting key developments in location-aware systems. Section III details the overview of the intelligent location-aware architecture. Section IV describes system design, including system architecture, data collection, and algorithm development. Section V presents experimental details of system deployment in a controlled environment. Section VI contains a discussion of experimental results. Section VII provides examples of two use-case scenarios. Section VIII presents potential future research developments in various domains. Finally, Section IX concludes the paper with future research directions.

II. Literature Review

In the rapidly evolving domain of mobile computing, the convergence of location-aware technologies with intelligent systems has driven a transformative shift towards more adaptive, responsive, and user-centric applications. This section reviews the significance and foundational principles underlying several important topics. Firstly, proximity authentication techniques are crucial for secure access to the proposed architecture by verifying the identities of users and authorizing their presence within specific geographic boundaries, providing an added layer of security beyond traditional credentials. This helps prevent unauthorized access to sensitive information and ensures that only approved individuals are able to interact with the architecture. Secondly, location-aware systems leverage positioning technologies to determine user locations, enabling a wide range of location-based services and applications to provide personalized services and enhance the user experience. Thirdly, context-aware recommendation systems utilize multidimensional contextual data in surrounding environments, including location, time of day, and user preferences, to deliver highly relevant personalized recommendations, significantly improving user engagement and satisfaction. Lastly, the capabilities of mobile computing environments require architectural considerations distinct from traditional computing paradigms. Effective integration of these topics into a reliable location-aware architecture is fundamental for delivering intelligent, context-sensitive solutions that can enable seamless, secure, and highly personalized location-aware services and adapt to the dynamic situations and needs of users in different mobile environments.

A. Proximity Authentication

Proximity authentication [12]–[14] is a method of authentication that uses the proximity of a physical device, such as a smartphone or smart card, to verify the identity of a user. This method is known as proximity authentication and is commonly used in security systems to ensure only authorized individuals can access restricted areas or information. Proximity authentication provides an additional layer of security beyond traditional methods like passwords or PINs. It is commonly employed to enhance the security of physical access control systems, such as buildings or secure areas, or to grant access to digital resources, such as online accounts or computer systems. The basic principle of proximity authentication is that a device, such as a smartphone or smart card, is associated with a specific user and can be used to verify their identity. When a user approaches a secure area or attempts to access a digital resource, the device is brought close to a reader or sensor that can communicate with it. The reader or sensor may use a variety of technologies to communicate with the device, such as RFID, NFC, Bluetooth, or Wi-Fi. The device then sends a signal to the reader or sensor, which verifies the identity of the user and grants or denies access appropriately.

B. Location-Aware Systems

Location-Aware Systems (LASs) [15], [16] are computing systems that detect and utilize location data to deliver useful information, services, or actions relevant to the geographical position of an individual or an object when they enter a vicinity area. These systems rely on positioning technologies, such as GPS, Wi-Fi, BLE, and RFID to accurately determine the location of a user or an object. Additionally, LASs often incorporate context awareness to sense and react based on their environment by interpreting and understanding situational factors like time, weather, and location. This approach provides users with more personalized and relevant services based on location and the surrounding environment. LASs are commonly used with mobile devices, such as smartphones and tablets. However, they can also be used with other devices, including vehicles, wearables, and Internet of Things (IoT) devices.

LASs can be classified into two types: Location-Based Services and context-aware systems. LBS are services that respond to queries containing the location information of a user. For example, a user may inquire about the nearest restaurant, the best route to a destination, or the weather forecast for their current location. Context-Aware Systems (CASs) are systems that adapt their behavior or functionality based on the context of the user or the device. Context is any information that can be used to characterize the situation of an entity, such as location, time, activity, preferences, or environmental conditions. CASs utilize various sensors and sources to collect and analyze context information, employing techniques to provide context-aware services, thereby adding intelligence, awareness, and smartness to the user experience.

The application of LASs, considering specific environments, involves a deep understanding of the unique attributes and requirements of each space. For instance, the system could provide users with real-time information about sales, product availability, or even navigation through the store in a retail setting. It could offer campus navigation, schedule management, and personalized learning resources in an educational environment. The system might assist with appointment scheduling, wayfinding, and patient-specific information for healthcare facilities. For hotels, it could provide personal services, room service ordering, and local attraction recommendations. Therefore, the LAS applications can be customized to meet the specific needs of different industries and businesses, making them flexible and adaptable.

In summary, LASs are becoming increasingly widespread and influential due to advancements in mobile devices, wireless technologies, and artificial intelligence. They have numerous applications across various domains, including navigation, tracking, monitoring, entertainment, education, health, and social networking. Therefore, modern LASs should leverage location information to make applications more personalized and provide tailored offers to users who reside in specific locations. These offers are delivered to users through personalized notifications, which are more effective than general-purpose push notifications. However, LASs also present challenges and risks like privacy, security, accuracy, reliability, and usability. Consequently, they require careful design, evaluation, and regulation to maximize their benefits and minimize their drawbacks.

C. Context-Aware Recommendation Systems

Context-Aware Recommendation Systems (CARs) [17]–[22] represent a significant advancement in personalized recommendation technologies, offering the potential to significantly enhance user satisfaction by providing more relevant and timely suggestions. The fundamental key of CARs is the concept of context, defined as any relevant information characterizing the situation of an entity, inferring a user or item, that influences the recommendation process. This includes temporal, spatial, social, and environmental factors. This means CARs incorporate diverse contextual information, such as time, weather conditions, location, and social connections, into the recommendation process, unlike traditional systems that rely only on user-item interactions. By considering these contextual factors, CARs can generate recommendations that are more personalized and tailored to the specific needs and preferences of each user, having the potential to greatly improve accuracy, adapt recommendations in real time, and significantly improve relevant suggestions.

CARS methodologies can be categorized into three primary approaches: pre-filtering, post-filtering, and contextual modeling. Pre-filtering involves filtering out irrelevant context before applying recommendation algorithms, such as removing items that are irrelevant to the current context or situation of users, while post-filtering modifies the recommendations generated from traditional algorithms based on contextual factors. Contextual modeling, the most integrated approach, directly incorporates contextual information into the recommendation model to generate more accurate and personalized recommendations, employing advanced machine learning techniques to capture the complex interactions between user, item, and context.

However, the challenges of implementing CARs include data sparsity and cold start problems, privacy concerns, and computational complexity. Future research in CARS is expected to address these challenges through advanced modeling techniques, privacy-preserving methodologies, and the exploration of cross-domain applications. The aim is to develop more sophisticated, efficient, and privacy-oriented systems that can adapt across various contexts and domains.

D. Mobile Computing

Mobile computing is the technology that enables users to transmit data, voice, or video from one device, such as a computer or any other wireless-enabled device, to another without the need for a physical link via cables. The importance of mobile computing has most recently involved remote work and study environments and has become a vital part of everyday life. It allows users to be in contact as long as a wireless connection is available. With the advancements in the wireless community, people can interact with others with handheld devices, such as smartphones, and become more productive without being linked to a stationary area with cabling and wires. Available devices supporting mobile computing ecosystems comprise smartphones, tablets, laptops, and wearable devices, such as smartwatches.

Mobile computing consists of three components, including mobile communication, mobile hardware, and mobile software. Mobile communication refers to exchanging data and voice through existing infrastructure, protocols, and data formats necessary for seamless connectivity. Mobile hardware refers to wireless devices necessary for network connections and operations. These devices are equipped with sensors, full-duplex data transmission, and the ability to operate on wireless networks like IR, Wi-Fi, and Bluetooth. Mobile software is usually the operating system and applications used on the hardware that allow the user to work or communicate while on the go. These operating systems provide various features, such as touchscreens, cellular connectivity, Bluetooth, Wi-Fi, GPS mobile navigation, cameras, video cameras, speech recognition, voice recorders, music players, and sensors. Mobile computing eliminates physical connections and enables users to access information and perform tasks anywhere.

Mobile computing impacts almost every aspect of modern life. Mobile devices with apps can enhance productivity, enabling people to collaborate, manage tasks, create documents, and access information wherever they go. Additionally, mobile computing has revolutionized industries, such as healthcare, transportation, and education by providing real-time data access and communication. This technology has also transformed the way individuals communicate, shop, and access entertainment, making it an integral part of daily life for many people. Furthermore, mobile computing has also opened up new opportunities for the development of new business models and opportunities for entrepreneurs. For example, mobile commerce has changed retail, empowering consumers to shop and make payments online. Mobile learning tools make education more accessible, interactive, and personalized. Moreover, mobile health technologies enable remote health monitoring, patient engagement, and telemedicine.

Future developments in mobile computing are expected to focus on emerging technologies, opening the way for more innovative applications and services. Integrating AI and ML is set to make mobile devices more intelligent, offering personalized experiences and predictive functionalities. The IoT will further expand the scope of mobile computing, connecting numerous devices, enabling intelligent environments, and allowing smartphones to interact seamlessly with smart homes, cars, and cities. Additionally, the advancements in reliable 5G networks will unlock faster data speeds and lower latency, enhancing the overall mobile computing experience.

III. Architecture Overview

This section proposes a conceptual overview of the intelligent location-aware architecture for mobile computing environments. This proposed architecture is designed to recognize and adapt to the unique characteristics of specific environments, thereby enhancing the user experience through tailored offers. The primary objective is to automatically provide individual mobile users in specific environments with local services, useful information, and personalized recommendations based on their current location and user context that meet the needs and preferences of the right mobile users at the right time in an appropriate environment. Meanwhile, mobile users can safely interact, access, and share information and services on their mobile devices when they enter the vicinity of a sandbox server. Similarly, mobile users can also communicate with each other directly within the same community over a local Wi-Fi network through specific communication protocols. The proposed architecture is a computing model that divides tasks between a sandbox server on the server-side and a mobile device on the client-side, allowing mobile users to directly and securely access and contribute information and services through their mobile devices connected to a local Wi-Fi network. The sandbox server is responsible for hosting, managing, executing, and contributing services and information to the mobile device. Mobile users communicate and interact with software applications within the sandbox server via their mobile devices when they enter the vicinity of the sandbox server. The mobile device provides a user interface through which a mobile user can initiate requests for content or services and display results from the sandbox server. On the other hand, the sandbox server allows service providers to participate by developing their own services and applications, which they can then upload to the sandbox server. By uploading these services and applications to the sandbox server, service providers can extend the functionality and capabilities available to mobile users. This enables a dynamic and customizable user experience and the ability to access a wider variety of content and services through mobile devices. An architectural overview is given in Figure 1 to illustrate the communication among a mobile device, a sandbox server, and service providers via a local Wi-Fi network.

The proposed architecture is designed and implemented to enable individual mobile users to communicate, collaborate, and share information and services through their mobile device within a specific environment. Meanwhile, a sandbox server attempts to interact with the mobile users to provide useful information, local services, and personalized recommendations based on their specific needs and preferences, utilizing the current situation and context of individual mobile users, including location data and user profiles. With an understanding of the specific context and dynamics of certain environments, the proposed architecture can offer more specific and contextually relevant information and services about individual user profiles and preferences than generic LBS platforms.

The proposed architecture also provides a solution to privacy and data protection issues by independently separating personal data stored on mobile devices from application services on a sandbox server. Each mobile user can fully control their own data, digital identity, and content, encouraging transparency through their mobile device. Additionally, mobile users can grant or revoke access to their personal data as needed and authorize external services, applications, or users to access it. Moreover, the communication is facilitated through a local Wi-Fi network, which allows for seamless and efficient data transfer between the mobile device, sandbox server, and service providers. This means that mobile users can securely and directly access relevant information and services in real time, without intermediaries. Additionally, the use of the local Wi-Fi network reduces reliance on cellular data without relying on a stable Internet connection, resulting in cost savings for mobile users. This ensures a smooth and personalized user experience while maintaining the privacy and security of user data.

In addition, with the emergence of Web 3.0 technology, the proposed architecture relies on the principles of decentralized technology to solve the problems of scalability, reliability, and fault tolerance, which are perfectly suitable for saving infrastructure and maintenance costs and improving the rapid exchange of data in a Peer-to-Peer (P2P) network. This means that a sandbox server can directly communicate, exchange, and share their services and information with each other by running across several sandbox servers in a decentralized network without having a controlling authority from a central server, making it more tolerant to single points of failure and less vulnerable to censorship or manipulation. The decentralized communication enables efficient data exchange and collaboration between sandbox servers, ultimately enhancing overall system performance and reliability. This decentralized approach also allows for higher scalability and flexibility, as new sandbox servers can easily and freely join and leave the network as needed at any time, which makes the system continuously and highly dynamic.

In a typical scenario, when a mobile user comes within the range of a sandbox server in a specific environment, the mobile user needs to authenticate before allowing them to establish the connection through a local Wi-Fi network. The mobile user initially verifies their identities by checking in to the sandbox server within the targeted environment based on Bluetooth proximity detection. The mobile user is then required to register with the sandbox server using a mobile number. After the sandbox server validates the mobile number and determines the mobile user is still within the allowed proximity, the sandbox server generates or retrieves a Wi-Fi password and utilizes an SMS gateway to send an SMS message with the Wi-Fi password to the registered mobile number of the mobile user. The mobile user receives the Wi-Fi password on their mobile device and enters the received Wi-Fi password to gain access to the resources, information, and services through the local Wi-Fi network. The mobile user can set up profiles with some basic information, including names, interests, skills, hobbies, and biographies, that allows the sandbox server to deeply understand the habits and preferences of the mobile user and accurately provide them with more personalized information and recommendations. Once the mobile user has completed their profile settings, the sandbox server interacts with the mobile user by offering local services, useful information, and personalized recommendations in the form of user interfaces and dialogues. In the situation where a new mobile user lacks historical and personal data, the proposed architecture initially suggests related information and services based on the current community, such as recently viewed items, frequently purchased items, best sellers, trending items, most viewed items, popular items, and featured items. When mobile users exist on the sandbox server, their personal and behavioral data are automatically deleted after 24 hours, according to a timestamp. This ensures that any potentially sensitive information is not stored for longer than necessary, further enhancing user privacy and data protection.

The proposed architecture consists of two main parts: (1) mobile devices; and (2) sandbox servers. This architecture plays important roles in serving a specific purpose, seamless communication, and engagement within a specific environment. Each part is explained in subsections, which describe how the sandbox server shares services and information and interacts with mobile devices.

A. Mobile Devices

Mobile devices serve as multifunctional tools that enhance various aspects of daily life, offering convenience, connectivity, and access to a wide range of services and resources. A mobile device includes the hardware and software components necessary

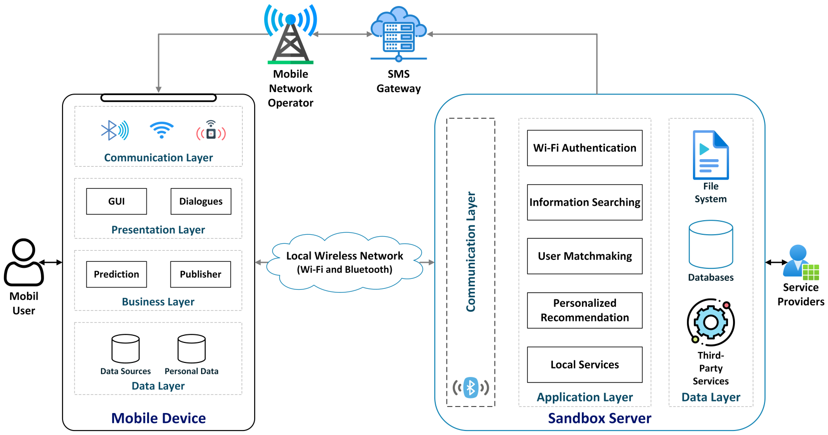

Fig. 1. An architectural overview of the communication among a mobile device, a sandbox server, and service providers via a local Wi-Fi network

for running applications. The hardware components of a mobile device typically include a processor, memory, storage, and various sensors present on the device, such as a GPS, Wi-Fi, BLE, and accelerometer. The software components consist of an operating system, application frameworks, and user interfaces. Additionally, it ensures that the mobile device is capable of securely connecting to a sandbox server and transferring data between them. Figure 2 shows a data flow diagram representing the flow of data through several processes, components, and interactions within a mobile device.

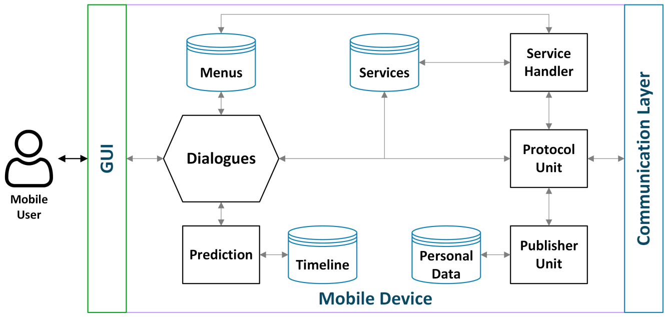

Fig. 2. A dataflow diagram of a mobile device

Mobile users obtain a set of services, information, and menus, depending on where they are, represented by local workflows through dialogues and user interfaces. These local workflows allow mobile users to easily navigate through various menus and access the services and information they need based on their location. As a whole, a mobile device can be divided into four primary layers:

1) Presentation Layer: This layer runs on a front-end device as a client-side system. A mobile user and a sandbox server interact and communicate primarily through a Graphical User Interface (GUI) with a compatible mobile application. A GUI consists of user interfaces and graphical elements, such as icons, buttons, and different menus, that allow mobile users to interact with them in dialogues. These menus can be presented within the GUI, including navigation menus, option settings, and other features accessible to mobile users. The dialogues are a communication process for exchanging data between a mobile user and a sandbox server that receives user input and displays information in response to user actions with dynamically updated content from the sandbox server, making them come alive with interactivity and adaptability. In the GUI, the visuals displayed in the user interface convey information relevant to the mobile user and actions they can take. Mobile users can usually interact with GUI elements by tapping a touch screen.

2) Business Layer: This layer is responsible for processing and managing the data received from the user interfaces. It handles the logic and rules of the application, ensuring that the data is validated and processed correctly. Additionally, the business layer communicates with the data layer to retrieve and update information from databases. This layer plays a crucial role in ensuring that the application functions properly and delivers the desired functionality to mobile users.

a) Service Handler: This is a component that is responsible for downloading and uploading services and menus from local databases on a sandbox server. This component ensures that the downloaded services and information are up-to-date by regularly checking for updates on the sandbox server. It also handles any errors or issues that may happen during the downloading or uploading process to ensure a smooth and uninterrupted flow of data.

b) Prediction Unit: This unit can automatically predict personalized, context-aware recommendations for possible next or similar Points of Interest (POIs), such as restaurants, shops, and attractions, based on user context triggered by the geographical position of a particular mobile user. It analyzes and learns sequential patterns from contextual data, historical timelines, and previous check-in data. These data are transformed into a multidimensional matrix as a tripartite graph-based data model called a location-based graph, storing users, POIs, and context nodes and establishing relationships between different POIs. A context node refers to contextual factors of a POI according to the individual mobile user, such as locations, location categories, times of the day, days of the week, seasons, and weather features, to enhance the accuracy of POI recommendations. Given such a location-based graph, advanced algorithms, including Personalized PageRank (PPR) and random walk techniques, are exploited to identify the most suitable POI recommendations for each mobile user. In particular, the context nodes are used to influence random walks to determine POIs that are relevant to the given contextual settings. Additionally, the location-based graph is constantly updated with new data to refine its understanding of user behaviors and preferences and adapt to changing environments. This ensures that the POI recommendations remain accurate and relevant, offering mobile users real-time suggestions that align with their current needs and situations.

c) Publisher Unit: This unit automates the creation of research profiles, interests, and achievements that highlight their work. It gathers personal information from public academic databases using web crawlers or Application Programming Interfaces (APIs) to collect bibliographic data, such as authors, affiliations, abstracts, and keywords. This data is stored in a personal database for further analysis. The unit then extracts relevant information for sharing with sandbox servers. For instance, if a mobile user enters a conference room, the unit might share their profile on a sandbox server, making it visible to others and suggesting relevant documents.

d) Protocol Unit: This unit contains several custom protocols that are designed and implemented for particular applications or services to fit their specific needs. The applications are used to interact with external systems and services. It is a set of rules, syntax, commands, and conventions that govern how different components within the application or service communicate and exchange information with each other.

3) Data Layer: This layer is responsible for storing and retrieving data from data sources. It provides the necessary functionality for the business layer to access and manipulate data. This layer contains the following data sources:

a) Personal Data: This database stores personal data that specifically identifies an individual mobile user, such as name, age, gender, education, contact details, expertise, biographies, and any other relevant data required for identification purposes. Additionally, it may include user preferences and settings to enhance personalized recommendations.

b) Timeline Data: This database stores data related to the activities and interactions of mobile users, such as their browsing history, check-in history, and search queries. These data are used to analyze user behavior and provide personalized POI suggestions and recommendations based on their interests and past actions. The timeline data also helps in understanding user patterns and trends, improving the overall user experience.

4) Communication Layer: This layer facilitates the secure exchange of data between a mobile device and a sandbox server. It acts as the first interface to manage the network connectivity of the mobile device, including incoming requests, validate them, and forward them to the appropriate components for processing, which enables a mobile user to directly and safely access information and services on the sandbox server through a local area network. There are a variety of communication technologies that enable a mobile device to establish connections with other devices or sandbox servers.

a) BLE: BLE is a short-range wireless transmission technology that is used for exchanging data between two different Bluetooth devices within a short distance using radio waves to communicate wirelessly. This Bluetooth technology is utilized for detecting the location information of mobile users, whether they are in radio coverage or not, and communicating with a sandbox server to request a Wi-Fi password before gaining access to resources or services on a local Wi-Fi network.

b) Wi-Fi: Wi-Fi is a wireless networking technology that allows mobile devices held by users to connect to a sandbox server. A Wi-Fi router assigns local IP addresses to connected devices, allowing them to communicate and exchange data with one another on a local Wi-Fi network.

In summary, the use of mobile devices allows mobile users to access a wide range of applications and services all the time. The software components ensure smooth operation and provide a user-friendly interface for seamless navigation and interaction. Additionally, the secure connection to a sandbox server ensures the protection of sensitive data during data transfer, enhancing user privacy and security.

B. Sandbox Servers

Sandbox servers refer to isolated and independent local servers that are locally set up in sandboxed environments, such as campuses, libraries, or restaurants. A sandbox server equipped with BLE adapters operates on the backend system as a server-side part. It can be physical or virtual machines (VMs) running on a powerful computer connected to a local area network within a specific environment. This means that the proposed architecture can also support multilayers of VMs to simultaneously run different modules on a single sandbox server. A sandbox server is responsible for receiving and handling requests from mobile users. It processes these requests and returns the corresponding responses to the mobile devices, while the mobile users wait for the responses and receive the responses delivered by the mobile devices. Additionally, a sandbox server acts as a bridge between the connected mobile devices and the Wi-Fi router, ensuring secure communication and authentication before granting access to the network. The sandbox server ensures that only authorized mobile devices can access the resources, information, or services on the local Wi-Fi network by requiring a valid Wi-Fi password. The data flow diagram in Figure 3 illustrates the various modules involved in the communication process, including the mobile devices, sandbox server, and Wi-Fi router, highlighting the flow of requests and responses between them. This diagram helps visualize how the sandbox server plays a crucial role in ensuring secure and authorized access to the local Wi-Fi network for mobile users.

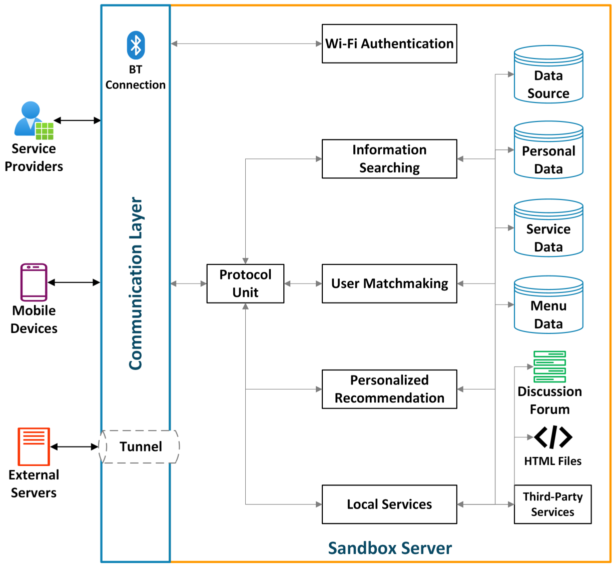

Fig. 3. A dataflow diagram of a sandbox server

A sandbox server is responsible for offering suitable local services, useful information, and personalized recommendations to an individual mobile user, depending on where the mobile users appear. The useful information refers to general information about the surrounding area, such as announcements, advertisements, weather conditions, local attractions, nearby restaurants, and upcoming events. The local services refer to the standard services that are available at the sandbox server, such as printing, scanning, games or puzzles, and sharing documents. Additionally, personalized recommendations can include suggestions for nearby activities, attractions, places to visit, or even local deals and discounts based on their profile and current situation. This allows mobile users to have a seamless and customized experience while accessing information and services within the environment of a station.

A mobile device held by a user can request any services and information from a particular environment over a local Wi-Fi network offered by a sandbox server and service providers. This can help reduce the spread of misinformation and disinformation, as mobile users can access credible and relevant information specific to their locations and contexts. The sandbox server also creates a dynamic ecosystem where service providers can continuously enhance and expand the range of information and services available on mobile devices. In addition, this sandbox server offers a direct connection to other sandbox servers via a tunneling network, so that data transmitted across this network is encrypted and encapsulated within a secure communication channel. Overall, a sandbox server contains three fundamental layers:

a) Application Layer: This layer runs on a sandbox server that manages the logic and functionality of the proposed architecture. It processes user requests, performs necessary calculations or operations, and communicates with the data layer to retrieve or store information. Additionally, it handles user authentication and security measures to protect sensitive information. This layer facilitates the integration of local services, personalized recommendations, and APIs, allowing for additional functionality and features to be incorporated into the proposed architecture. This application layer also communicates with the presentation layer to receive user input and deliver appropriate responses.

b) Data Layer: This layer stores and manages all relevant data exchanged between a sandbox server and mobile devices in databases. This includes user profiles, preferences, and any other data necessary for the functionalities of the proposed architecture. The data layer ensures that all user data is securely stored and can be accessed when needed. It also handles data synchronization between the sandbox server and mobile devices, ensuring that the most up-to-date information is available to mobile users at all times. Moreover, these databases can store data both temporarily and permanently. Sensitive data, in particular, is automatically deleted after 24 hours, according to a timestamp. This ensures that any potentially sensitive information is not stored longer than necessary, further enhancing user privacy and data protection. This feature provides extra security for the databases, ensuring that unauthorized users cannot access sensitive data for an extended period of time.

c) Communication Layer: This layer handles incoming requests from mobile users and manages the transmission of data between a sandbox server and mobile devices, ensuring that data is returned to mobile users in a timely manner. It provides secure and efficient communication via a local Wi-Fi network by implementing protocols that ensure all communications are encrypted, protected from unauthorized access, and properly protected during transmission. These protocols include Hypertext Transfer Protocol (HTTP), Hypertext Transfer Protocol Secure (HTTPS), and WebSocket, which are utilized to send requests from the mobile user to the sandbox server and responses from the sandbox server to the mobile user. The WebSocket protocol especially provides a powerful, persistent connection for establishing bidirectional, real-time, and event-based communication channels between a mobile user, typically a web browser, and a sandbox server using WebSockets over a single Transmission Control Protocol (TCP) connection. It allows for frequent data updates and instant interaction. With this connection, the sandbox server can push data to the mobile user in real time while simultaneously receiving data from the mobile user. It eliminates the need for continuous polling or long-polling techniques commonly used in traditional HTTP-based communication. As a result, it significantly achieves scalability by reducing the overhead of HTTP requests and responses, facilitating low-latency data transfer, and improving the efficiency and speed of data transmission. This capability is particularly beneficial for applications that require real-time updates, such as chat applications, notification platforms, and multiplayer games. Additionally, it facilitates real-time synchronization of data between the mobile device and the sandbox server, ensuring that both systems stay updated with the latest information.

The use of sandbox servers can increase accessibility for mobile users by providing them faster access to information, more control over the data stored on them, and less reliance on costly and unreliable Internet connections. Similarly, sandbox servers can keep local networks safe, which are still invisible from the Internet. Sandbox servers can also improve security because they are less likely to be attacked or have their data stolen than commercial or global servers.

C. Key Features and Functionalities

The proposed architecture utilizes diverse technologies, including mobile computing, AI, ML, and location-aware technology. The aim is to offer helpful information, tailored recommendations, enhanced services, improved decision-making capabilities, and more effective operations, which are essential in various sectors, including education, retail, healthcare, and urban planning. The key features and functionalities of the proposed architecture are explored, highlighting deeper insights into their significance, capabilities, and applications.

1) Wi-Fi Authentication Mechanisms: This is a security process to verify the identity of a mobile user attempting to access a local Wi-Fi network. This authentication process supports monitoring and controlling network access, ensuring only authorized mobile users can connect to the network. Additionally, this process also helps in enhancing overall network security and protecting sensitive resources and information from potential cyber threats.

2) Data Ownership: Mobile users have greater control over their data. They can determine who has access to their information and how it is used, providing a sense of privacy and security. This ownership also empowers users to make informed decisions about sharing their data with others, third parties, or organizations.

3) User Profile Integration: The proposed architecture integrates detailed user profiles, which include preferences, interests, hobbies, skills, and biographies, to offer highly personalized recommendations. This user-centric approach ensures that the information and services provided are relevant and engaging to each individual mobile user. By incorporating user profiles into the proposed architecture, individuals can receive tailored content and suggestions that align with their unique preferences and profiles.

4) Search and Filter: Mobile users can efficiently search for specific files or information stored in databases by entering search queries or asking questions in natural language that match their needs. The search results are provided in natural language responses that are contextually appropriate, relevant, and accurate for mobile users. This search approach involves a deeper analysis of intent, context, and conceptual meanings rather than just matching keywords. This feature enables mobile users to easily find content without prior knowledge of relevant keywords.

5) Personalized Recommendation: The proposed architecture recognizes the physical location of the mobile user within the predefined environment and delivers location-specific services, items, and information tailored to that location. Mobile users can also receive real-time content, advertisements, local news, and relevant updates based on their location, preferences, and profiles. By understanding the context of the surroundings of users, the architecture can offer more targeted and valuable suggestions for a seamless mobile experience. Additionally, this personalized recommendation feature can suggest nearby POIs that may interest the mobile user.

6) Social Interactions: Mobile users can discover events, friends, or interests nearby based on their location and preferences, creating opportunities for new connections and experiences. Additionally, mobile users can share their own experiences and recommendations with others, supporting social connections and community building.

7) Push and Pull Interaction Mechanisms: Mobile users can interact with the proposed architecture through both push and pull mechanisms. The proposed architecture can proactively push information to mobile users based on their current location and profile, such as sending alerts for nearby events they might like or suggesting new friends based on shared interests. On the other hand, mobile users can pull information from the proposed architecture based on their immediate needs or interests, such as searching for specific events happening in their area, looking for friends with similar hobbies, or querying for nearby dining alternatives. This bidirectional interaction allows for a dynamic and personalized user experience.

8) Customer Engagement: By providing mobile users with personalized information and recommendations, the proposed architecture encourages active engagement with the environment. It opens possibilities for mobile users to discover new opportunities, experiences, or interests that align with their preferences, enhancing their overall experience in the predefined environment.

9) Scalability and Adaptability: The proposed architecture is designed to be scalable to different environments and adaptable to varying user preferences, profiles, and needs. This allows the proposed architecture to naturally adapt to changes in user behaviors or new features without compromising performance or accuracy. Additionally, the proposed architecture can be easily integrated with existing technologies and platforms for seamless implementation across different settings.

To summarize, the proposed architecture provides advanced features and functionalities that make it highly flexible and adaptable to various applications. With its ability to communicate securely, maintain deep contextual awareness, conduct semantic meaning analysis, perform predictive analytics, and integrate seamlessly, the proposed architecture offers personalized, dynamic services that can improve efficiency, safety, and the overall user experience in the digital era.

IV. System Design

The system design of a sandbox server comprises various software modules that make up the server-side part and provide functionality for mobile users. These software modules are responsible for managing user requests, processing data, and ensuring the overall functionality of the sandbox server. Additionally, it involves setting up encryption protocols to further enhance security measures and protect sensitive information stored on the sandbox server. This section delves deep into the detailed description of each software module and its role within the sandbox server ecosystem.

A. Wi-Fi Authentication

When setting up a local Wi-Fi network, a robust Wi-Fi authentication mechanism is a fundamental aspect of network security that acts as the first layer of defense for the local Wi-Fi network. It involves verifying the identity of a user before allowing it to connect to the local Wi-Fi network. This mechanism is essential to protecting network access and user-sensitive data. It also provides security for data communications between mobile devices on the local Wi-Fi network so that authorized individuals can only use the network. This Wi-Fi authentication module is accomplished through Wi-Fi authentication mechanisms based on the presence or proximity of mobile users. Proximity-based Wi-Fi authentication uses location-based BLE technology, enabling mobile users to authenticate using their mobile devices via Bluetooth connection. This module adds an extra layer of security by ensuring that only mobile users within a certain range can access the local Wi-Fi network. This means that it uses the distance between a mobile device and a sandbox server as a key measurement to verify the identity and determine that the mobile user is in close proximity to the sandbox server on a trusted mobile device. This is a security process that acts as an interface in the middle between a Wi-Fi router and mobile devices. A mobile user needs to make a registration or check in by providing a phone number via Bluetooth connection to receive a Wi-Fi password according to the registered mobile phone via a Short Message Service (SMS). It ensures that the owner of the phone has given permission for mobile users to access the local Wi-Fi network. By implementing this authentication approach, the local Wi-Fi network can prevent unauthorized access and maintain a secure environment for its mobile users. A diagram for the Wi-Fi authentication module is illustrated in Figure 4.

From Figure 4, the main idea is to verify the identities of mobile users when their mobile device is in close proximity to a Bluetooth adapter and then send Wi-Fi passwords directly to the registered mobile number of the mobile users via SMS. SMS refers to a short message containing a Wi-Fi password that is sent to the mobile phone of the mobile user who initiated the request. The mobile users then gain access to local resources, information, and services. This method not only simplifies the authentication process for mobile users but also reduces the risk of unauthorized access to the local Wi-Fi network. By utilizing Bluetooth technology and SMS verification, the system ensures a secure and seamless connection for authorized mobile users.

The authentication process starts with a mobile user holding a Bluetooth-enabled mobile device that attempts to connect to

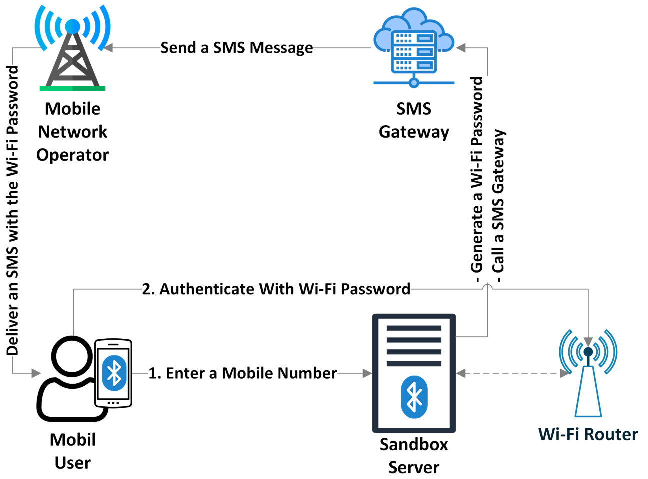

Fig. 4. A dataflow diagram of the Wi-Fi authentication module

information and services on a local Wi-Fi network. Before accessing the local Wi-Fi network, the mobile user needs to use the Bluetooth-enabled mobile device to scan for advertising signals broadcast from available Bluetooth adapters. Bluetooth adapters act as devices to advertise and wait for connections, which accept an incoming connection request after advertising. If the mobile device detects a Bluetooth adapter and determines the mobile user is in proximity, the Bluetooth adapter accepts the pairing and bonding processes. Once the pairing and bonding processes are completed, the mobile device can establish a secure connection and exchange information with a sandbox server on a Radio Frequency Communication (RFCOMM) channel. The mobile user is required to provide a mobile number. After the sandbox server determines the mobile user is still within the allowed proximity, the sandbox server generates or retrieves a Wi-Fi password and utilizes an SMS gateway to send the Wi-Fi password to the registered mobile number of the mobile user. The mobile user receives the Wi-Fi password on their mobile device. To gain local Wi-Fi network access, the mobile user must enter the received Wi-Fi password on the screen of their mobile device to prove their identity. Once the Wi-Fi password is entered correctly, the mobile user will be granted access to the local Wi-Fi network. By doing this, it makes sure that the owner of the phone has verified that the mobile user is accessing the local Wi-Fi network. The Wi-Fi router establishes and acknowledges the connection if the entered Wi-Fi password is correct. Suppose at any point the mobile user is not authenticated or is not in the allowed proximity. In that case, the sandbox server sends an error message to their mobile device and displays an error or a retry message to the mobile users. This scheme indicates that it is highly convenient, safe, and smooth for mobile users, thereby enhancing their trust and confidence in using resources, information, and services on the local Wi-Fi network.

An SMS gateway refers to a server that can send or receive text messages in the form of SMS transmissions between local and global telecommunications networks, primarily between the Internet and mobile networks. An existing software application acts as a sender that can send SMS text messages through SMS APIs or web portals to an SMS gateway. SMS APIs refer to a software integration interface responsible for connecting between telecommunications carrier networks and the Internet to easily send and receive SMS text messages. The SMS gateway receives and translates the SMS message into the appropriate format from the format used on the Internet, such as email or HTTP protocol, to a format compatible with mobile networks, such as Code Division Multiple Access (CDMA), Global System for Mobiles (GSM), and Long-Term Evolution (LTE). This ensures that SMS messages can be sent and received between different types of networks and devices seamlessly. After that, the SMS gateway interacts with the correct mobile network operator to send the SMS message to the best carrier, ensuring the message is delivered to the recipient's mobile device. Finally, the mobile network operator responds back with updates about the message delivery status to the original sender.

In summary, Wi-Fi authentication and encryption are used in pairs to prevent local Wi-Fi networks from unauthorized and malicious access attempts and secure wireless transmissions. This Wi-Fi authentication based on Bluetooth proximity acts as an interface between a Wi-Fi router and mobile devices. It enables only the phone number owner to receive a Wi-Fi password, allowing them to log in to a local Wi-Fi network and verify their identity using a received password via an SMS message. This makes it difficult for attackers to obtain unauthorized access to data and resources or to steal user credentials. It differs from traditional password authentication, which may continue to be useful for attackers with stolen credentials. Additionally, it helps prevent blocking all traffic except for authentication traffic, which could enhance the security and privacy capabilities of the local Wi-Fi network and improve the user experience.

B. Information Searching

The information searching module offers search capabilities with Large Language Models (LLMs), such as GPT, LLaMA, Claude, Gemini, and PaLM, enabling mobile users to locate specific information or documents from various data sources present on a sandbox server to meet their needs. This module allows mobile users to enter queries or ask complex questions in human language and receive relevant and contextual responses in a way that is quite natural and conversational, which could enhance the search experience and user satisfaction. The key components and processes involved in the software module are illustrated in Figure 5.

The primary goal is to provide mobile users with more accurate and contextually relevant responses, focusing on fetching the most current relevant information and understanding the intent, semantic, and contextual meaning of search queries from mobile users. This is different from traditional search engines that return a list of documents based on exact matches of words or phrases. To achieve this, the module combines three fundamental aspects of advanced natural language processing algorithms and machine learning techniques to enhance the relevance, quality, and reliability of the final generated responses: (1) retrieval-based component; (2) generation-based component; and (3) index-based component. The following subsection explains each component in detail.

From Figure 5, when a mobile user makes a query or question, the retrieval-based component first retrieves a set of relevant contexts corresponding to the given search query from the vector database. Once the relevant contexts are retrieved, they are concatenated along with the original user query and transformed into a prompt. Finally, the prompt is sent to the LLM for text

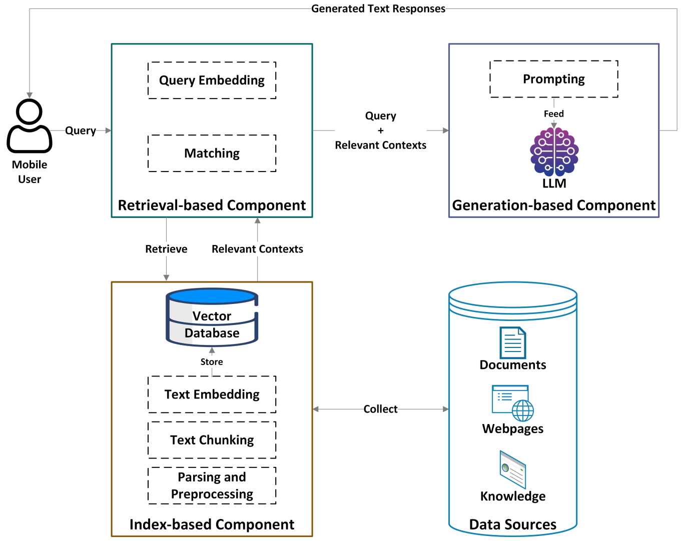

Fig. 5. A diagram of the information searching module

response generation. The LLM utilizes its knowledge, contextual understanding, and language generation capabilities to summarize and extract important information within the given relevant contexts to generate responses that are contextually appropriate, relevant, accurate, and coherent for the mobile user.

1) Retrieval-based Component: The retrieval-based component adapts the core idea of Retrieval Augmented Generation (RAG) to retrieve coherent and contextually relevant pieces of text with facts from data sources and knowledge bases based on a given search query or input. The main goal of this component is to improve the performance and flexibility of the generation-based component by adding new information and domain-specific data on top of the initial training data of models that were previously trained on a large dataset. This is useful for the generation-based component because it can reduce the risks of the generation of false or misleading information, called hallucination, enabling it to produce more appropriate, accurate, and reliable outputs. The component starts receiving a search query or question given by a mobile user and then utilizes the query embedding process to convert the given query into a numerical representation as a query vector. It achieves this by using an embedding language model, such as BERT, GPT, or RoBERTa, that captures the semantic meaning behind the given query. The query vector is typically generated using the same embedding language technique used to create the indexed vectors. The matching process is then employed to find the nearest neighbors of a query vector from the indexed vectors stored in a vector database using an Approximate Nearest Neighbors (ANN) algorithm. The idea behind ANN algorithms is to efficiently search for similar vectors in a high-dimensional space without computing distances between a query vector and all indexed vectors in the vector database, so that it only computes distances between the query vector and the small number of candidate vectors around it. Finally, the similarity scores between the query vector and the indexed vectors are measured using a distance metric, such as cosine similarity or Euclidean distance. The top-k most similar vectors are returned as the relevant contexts ranked based on their similarity scores.

2) Generation-based Component: The generation-based component uses an LLM that has been trained on a huge number of general domain corpora. Its goal is to generate text responses that are more contextually appropriate and accurate based on the relevant contexts provided by the retrieval-based component. However, traditional LLMs generate text based on patterns learned from a large public corpora, but they may not always have access to domain-specific data or current information. To address the limitations of general-purpose LLMs, this component combines the relevant contexts as additional contexts retrieved from the retrieval-based component with the original query given by a mobile user as a prompt. The prompt is augmented using the prompting process through prompt templates and instructions to ensure that the generated text is tailored to the specific domain or topic of interest. The new augmented prompt is then fed to the LLM to generate contextually relevant responses as the final output for the mobile user. This final output can take various forms, depending on purposes, such as a natural language response, a summary, a translation, or any other text-based output.

3) Index-based Component: The index-based component is a crucial initial step in data preparation that occurs in the offline stage and involves several processes. It begins with fetching textual data sources stored on a sandbox server. These textual data sources include databases, knowledge bases, and a collection of documents, such as books, scientific articles, reports, and webpages, containing various file formats, such as Portable Document Format (PDF), Hypertext Markup Language (HTML), Text File Document (TXT), Microsoft Word Document (DOCX), and Power Point Presentation (PPTX). These textual data or documents are then converted, parsed, and preprocessed into standardized plain texts. To fit within the context limitations of language models, the text chunking process is used to split the plain texts into small pieces of text known as text chunks. A text chunk usually contains 256 or 512 tokens, making the text chunk more manageable for further processing. This chunking process is essential for efficiently accessing and analyzing specific sections of text. These text chunks are subsequently transformed into embedding vectors, which are numerical representations that capture their semantic meaning in a high-dimensional space using an embedding language model, a process known as text embedding. This facilitates similarity comparisons with the embedded queries during the retrieval-based component. Finally, an indexed vector dataset is created to keep these text chunks and their generated embedding vector as key-value pairs and stored in a vector database, which allows for efficient and scalable search capabilities.

In summary, this information searching module has a remarkable capacity for understanding natural language to accurately interpret user queries and provide relevant response results that are more contextual and informative. By augmenting LLMs with external knowledge sources, this module can improve the quality of responses generated by LLMs, making these language models more reliable. Moreover, this module not only enhances the quality and relevance of LLM outputs but also ensures that LLMs can adapt and stay up-to-date with the latest information, facts, and trends that may change over time. This reduces the chance of providing outdated or inaccurate information and enables explainability and transparency. This represents a significant improvement in making LLMs more practical and useful for downstream applications, such as chatbots, search engines, or recommender systems, where recent and domain-specific knowledge is crucial.

C. User Matchmaking

The user matchmaking module aims to help mobile users discover potential users who have similar interests, preferences, and traits within a virtual community based on PPR and semantic analysis. This module considers only textual user profiles instead of mutual friends or social graph criteria. A set of user profiles is transformed into a graph model for efficiently discovering hidden connections between users. The semantic relationship between two user profiles is then estimated using word and sentence embeddings. Moreover, the personalization feature of the graph allows the module to specify the number of neighboring nodes surrounding a source node. By incorporating both embedding models and personalized graph analytics, it can capture complex semantic information, individual priorities, and high-order user relationships, making the matchmaking process more accurate. The processing pipeline of this module is illustrated in Figure 6.

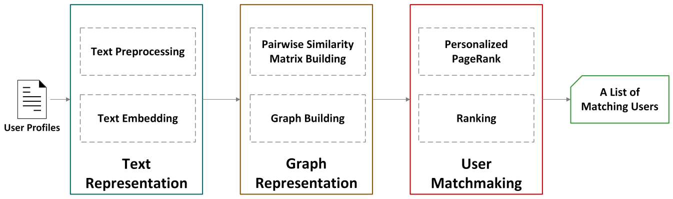

Fig. 6. The pipeline of the user matchmaking module

The pipeline starts by preprocessing each user profile. This involves cleaning the textual data, removing stop words, and stemming words. Each text attribute of a user profile is then converted into a dense vector representation using embedding models. The cosine similarity score is calculated for each attribute pair of user profiles. These scores are then averaged to get an aggregated similarity score. The score is inserted into a similarity matrix to construct a weighted graph, where nodes represent users and edges represent their relationships. The weight of an edge can be an aggregated similarity score, which indicates the strength of the relationship between two users. The PPR method is then used to identify influential users based on a target user within the weighted graph and to generate a list of matching users who have similar profiles. This consists of three main components: (1) text representation; (2) graph representation; and (3) user matchmaking. The following subsection explains each component in detail.

1) Representation Component: The text representation component preprocesses user profiles and converts them into high-dimensional vector representations. These vector representations enable efficient comparison and matching of user profiles based on their textual information, which captures the semantic meaning of individual words and texts. This component begins with text preprocessing, aiming to clean and standardize unstructured textual profiles into a suitable format through tokenization, lowercase conversion, removing stopwords, punctuation, and special characters, stemming, and lemmatization. Additionally, the component employs text embedding to convert each text attribute of a user profile into a dense vector representation using two embedding models. The first embedding model is term-based embedding. This considers the text attributes of a user profile, including interests, hobbies, occupations, and skills. Each text attribute is mapped into the word vectors of all words in the attribute using Word2Vec, an existing pre-trained word embedding model with 300-dimensional word vectors. These word vectors are then averaged to generate a single vector, representing the attribute. The second model is context-based embedding. This focuses on the biography attribute in a user profile. The biography attribute is directly converted into a single vector representation using a pre-trained Bidirectional Encoder Representations from Transformers (BERT), a transformer-based model. Instead of aggregating word vectors, this model considers the entire text as input and generates comprehensive embeddings that capture the overall semantic content. Finally, the next component uses the embedded vector representations to compute the semantic similarity scores between user profiles.

2) Graph Representation Component: The graph representation component utilizes a pairwise similarity matrix to construct a weighted graph representing semantic relationships between users. The pairwise similarity matrix is built by calculating the cosine similarity between every pair of user profiles based on their vector representations and inserting them into the matrix. The cosine similarity calculation calculates the semantic similarity score between two single vectors for every attribute pair. This calculation is performed for each attribute pair in the user profiles, resulting in a similarity score for each pair. To combine different attributes of a user profile, an aggregated similarity score is calculated for two user profiles by taking a simple average of all the cosine similarity scores across their multiple attributes. The aggregated similarity score of two users is inserted into the pairwise similarity matrix. Finally, the weighted graph is constructed from the pairwise similarity matrix. Each node in the graph represents a user. If the aggregated similarity score between two nodes exceeds a certain threshold set at 0.70, they are connected with an edge. The weight of the edge is determined by the aggregated similarity score, which indicates the strength of the relationship between them. A high score refers to a strong relationship, and a weak relationship has a low score.

3) User Matchmaking Component: The user matchmaking module utilizes the PPR algorithm to identify the most influential users based on their similarity to a target user within a weighted graph. The PPR algorithm takes into account the relevance of each connection and similarity score, ensuring that users with more meaningful connections are ranked higher. The PPR algorithm modifies the original PageRank algorithm slightly to produce personalized matchmaking for users. Instead of moving to any random node, the algorithm biases the random walk towards a specific starting node or a set of nodes based on a personalization vector. This provides a set of nodes that are reachable from the starting node. The PPR algorithm starts by setting the entry corresponding to the target user to 1 in the starting vector. It then updates the vector iteratively until convergence. The algorithm assigns higher scores to users who are reachable from the target user. After convergence, the module sorts the user indices based on their PageRank scores for the target user in descending order. The top-k users are the users who are most similar to the target user, where k is a threshold or specific number.

In summary, by incorporating the semantic similarity of users into textual data, it is possible to obtain an in-depth understanding of user preferences, interests, and traits. This module can enhance the accuracy and effectiveness of personalized user matchmaking by considering not only social structures and behaviors but also the content shared by users. Additionally, integrating personalized graph-based approaches into the matchmaking process can further improve the quality of personalized user matchmaking by leveraging the semantic relationships between users within the graph.

D. Personalized Recommendation

The personalized recommendation module aims to personalize the textual profiles of mobile users by providing highly relevant and useful recommendations about items, information, or people. This module leverages the underlying idea of advanced Information Retrieval (IR) techniques by incorporating both semantic-based and keyword-based search capabilities as a hybrid search to enhance the accuracy, personalization, and relevance of recommendation results. It is a powerful way of analyzing item descriptions and textual profiles that require the capabilities of complex semantic matching, contextual relevance assessments, and simple keyword matching, such as product names or serial numbers. The semantic-based search component utilizes NLP and ML techniques to understand the contextual and semantic meanings of the text, while the keyword-based search component focuses on matching exact terms or phrases. The diagram of the personalized recommendation module illustrates how the hybrid search approach combines the strengths of semantic-based and keyword-based searches in Figure 7.

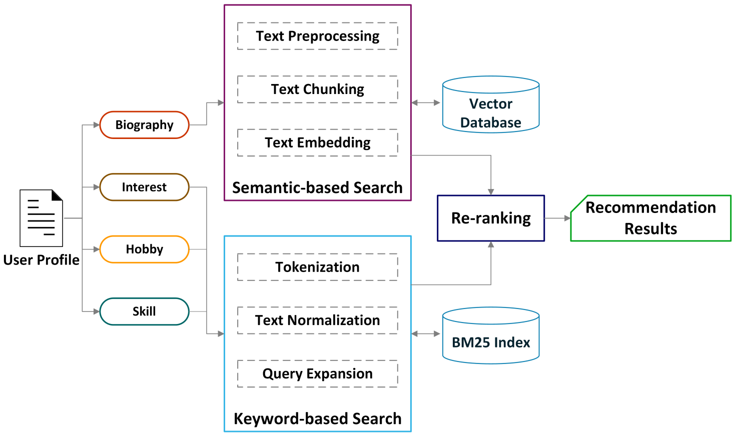

Fig. 7. The diagram of the personalized recommendation module

Based on the user profile of a particular mobile user, this module is used to retrieve relevant semantic and contextual information about items or people from various data sources to make personalized recommendations. These data sources can include documents, books, websites, scientific articles, or any other repository of information relevant to a sandbox server. The text attributes of a user profile are preprocessed using components of semantic-based and keyword-based search to transform the text attributes into standard forms for further analysis. The semantic-based search component identifies the most contextually relevant information and items that closely match a given textual attribute from a vector database containing embedding vectors. The keyword-based search component discovers relevant information and items based on specific keywords or phrases in a BM25 index. The sparse vector is suitable for discovering information based on specific keywords, while the dense vector is suitable for retrieving information based on contextual and semantic similarity. These two components work together to provide a comprehensive analysis of the text attributes to enhance the functionality and accuracy of the recommendation process. Each component returns a list of search results based on their similarity scores. Subsequently, the original search results are re-ranked using the Reciprocal Rank Fusion (RRF) algorithm to refine the recommendation results, ensuring that the final recommendation results are tailored to the specific preferences and profiles of individual mobile users. The personalized recommendation module comprises three main components: (1) semantic-based search component; (2) keyword-based search component; and (3) re-ranking component. Each of these components plays a crucial role in ensuring that mobile users receive personalized recommendations that are relevant and engaging.

1) Semantic-based Search Component: The semantic search component focuses on understanding the semantic, contextual, and intent meanings of a biography attribute as a textual description of an individual mobile user and providing contextually relevant results. First, a biography attribute is preprocessed and split into several chunks. These chunks are then encoded into vector embeddings as dense vector representations using an embedding language model. Second, these vector embeddings are used to calculate the semantic score similarity between the biography attribute and other information or items stored in a vector database based on cosine similarity. Third, the semantic search component returns a list of search results that match the biography attribute. Finally, the list of search results with similarity scores is passed to the re-ranking component for further refinement.

2) Keyword-based Search Component: The keyword search component analysis leverages the textual attributes of a user profile, which refers to the collection of information about an individual mobile user, including interests, hobbies, and skills, to make personalized recommendations. Each attribute is cleaned and preprocessed, a process called text normalization. Text normalization involves transforming text into a standard form, including operations like tokenization, stopword removal, and stemming. The query expansion process is then performed to reformulate each attribute query to include synonyms, different lexical forms of words, and other related terms, allowing for improved search relevance and quality by finding relevant information and items that use different terms to express the same concept. These items do not necessarily contain the exact terms used in the search query but are still relevant. Next, each attribute is transformed into an embedding vector as a sparse vector representation to calculate the similarity score between the attribute vector and a BM25 index created using the BM25 ranking algorithm. The retrieval process then identifies the most relevant information and items and returns a list of search results that match the attribute. Finally, the list of search results with similarity scores is passed to the re-ranking component for further refinement.

3) Re-ranking Component: The semantic-based and keyword-based search components return a separate set of search results with different similarity scores, usually a list of search results sorted by their relevance scores. Instead of randomly ordering the separate sets of search results, they are combined and re-ranked using the RRF algorithm for the final recommendation results. The RRF algorithm combines results from different retrieval systems based on document positions or ranks. This re-ranking component ensures that the final recommendation results are tailored to the profile and preferences of individual mobile users, providing a more personalized and accurate experience.

In summary, the combination of semantic-based and keyword-based searches has an exceptional ability to comprehend the lexical, semantic, and contextual meanings of user profiles in natural language. This can significantly enhance the quality of generating recommendation results in terms of accuracy, relevance, and personalization, ultimately leading to improved user satisfaction and engagement. Importantly, this approach can also address the cold start problem for new users without solely relying on historical interactions.

E. Local Services

With the local service module, mobile users have the convenience of accessing and managing various services seamlessly within a sandbox server environment. This module provides adaptation information and services related to the environment to mobile users, depending on their current location. Figure 8 shows the data flow and actions within the local service module, illustrating how mobile users can easily navigate and access relevant services based on their location.

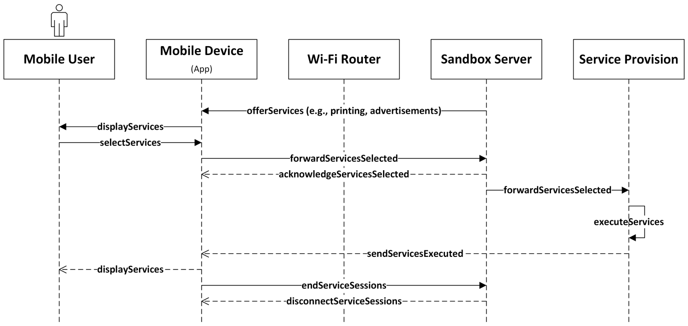

Fig. 8. A sequence diagram of the local service module

Local services are a set of basic requirements available in an environment to meet the expectations of mobile users. Ideally, the local services include printing services, scanning services, document sharing, discussion forums, chat, local web pages, announcements, games or puzzles, advertisements, campaigns, and promotions. These local services are designed to enhance the user experience and provide convenience for individuals. By offering various options, mobile users can efficiently utilize their current location to access necessary services, resources, and information.

From Figure 8, the diagram starts when a mobile user enters the range of a sandbox server and a local Wi-Fi network connection is established. The sandbox server detects that a new device has connected to the local Wi-Fi network. It then triggers a process to send a welcome message or a local service menu to the mobile device. This could be implemented as a push notification, an automatic redirect to a web page, or an app notification. The mobile device displays the message and the list of local services to the mobile user. The mobile user reviews and selects from the list of available local services provided by the sandbox server through their mobile device. The list of local services includes printing services, scanning services, discussion forums, announcements, and advertisements. The mobile user sends a service request message to the mobile device, indicating their desired local services. The mobile device acts as an intermediate device that forwards the service request to the sandbox server, which is responsible for handling the request. The sandbox server processes the service request and sends a confirmation message back to the mobile device. The mobile user then receives a notification on their mobile device, confirming that their request has been successfully submitted for processing. Meanwhile, the service provision executes the requested services on the sandbox server. For printing and scanning, it could mean sending a job to a local printer or scanner. For informational services like forums, announcements, and advertisements, it could involve maintaining an active session where the mobile user can interact with the content. Once the requested services are processed, the sandbox server sends a response back to the mobile device and displays information on how to access the requested services to the mobile users. This seamless communication between the mobile user and the sandbox server allows for efficient access to the various services offered. After the mobile user has finished using the local services, there might be a process for terminating the session, which can be initiated by the mobile user, the sandbox server, or by disconnecting from the local Wi-Fi network.

In conclusion, these software modules work together to develop a secure and efficient environment for mobile users to interact within a sandbox server. The user authentication module ensures that only authorized mobile users can access resources on the sandbox server, preventing unauthorized access and potential security breaches. The information searching module allows for efficient storage and retrieval of data within the sandbox server, ensuring rapid access to information and the satisfaction of search results. Additionally, the user matchmaking module facilitates connections between users based on their preferences or other criteria. The personalized recommendation module provides mobile users with tailored suggestions and recommendations based on their current location, user profiles, and preferences. Finally, the local service modules enable mobile users to control and play local services in the environment of the sandbox server, including any technology or equipment available.

V. Experiments

In this section, a series of experiments is conducted to evaluate the robustness, scalability, and effectiveness of the proposed architecture. The objective is to ensure that the proposed architecture can perform well and meet the needs of mobile users in different environments. The experiments involve testing the ability of the proposed architecture to accurately detect user locations, support a high number of concurrent users, handle high data traffic, maintain web pages under different levels of concurrent users, and handle the response time of APIs for varying request rates. The results of these experiments provide valuable insights into the capabilities of the proposed architecture and its potential to perform further improvements to enhance its overall performance and user satisfaction. The experiments are structured into experimental strategies and experimental settings. Generally, the conducted experiments aim to answer the following Research Questions (RQs):

- **RQ1**: How is the estimated range and reachability of Bluetooth signals between a mobile user and a sandbox server?
- **RQ2**: How does the proposed architecture accurately identify the location of a mobile user?
- **RQ3**: Can the web pages hosted on the proposed architecture handle a large number of concurrent users?
- **RQ4**: What maximum user load can web pages handle before the response time exceeds an acceptable threshold?
- **RQ5**: Can the APIs hosted on the proposed architecture support a certain number of requests per second?

- **RQ6**: What is the maximum request rate that APIs can handle before the response time exceeds an acceptable threshold?

The detailed experimental procedures are shown in the following subsections:

A. Experimental Strategies

The experimental strategies aim to make realistic simulations from different points of view so that the proposed architecture can be tested for its performance. These tests are divided into four categories: (1) Bluetooth signal testing; (2) location determination testing; (3) web page load testing; and (4) API load testing. The following strategies are explained below:

1) Bluetooth Signal Testing: This strategy measures the strength and quality of Bluetooth signals between a mobile device and a Bluetooth adapter. The Bluetooth signal testing aims to estimate the range of Bluetooth signals and how they are reachable, ensure that the Bluetooth signals are stable and reliable, and support the required data transfer rate for given use cases. In the context of Bluetooth, the Received Signal Strength Indicator (RSSI) is a key metric used to estimate the distance between two Bluetooth-enabled devices. RSSI values captured from the signals of two Bluetooth-enabled devices are used to indicate the strength of a Bluetooth signal at a specific location. Generally, the common range of RSSI values is between -100 dBm and -20 dBm. An RSSI value of -30 dBm indicates a strong Bluetooth signal, while an RSSI value of -90 dBm indicates a weak signal. In addition, a reasonably acceptable value is -70 dBm to -80 dBm.

2) Location Determination Testing: This strategy involves conducting tests to assess the accuracy and reliability of Bluetooth signals in determining the precise location of a mobile device in terms of distance from the actual location in a specific area. This testing involves comparing the actual distance between a mobile device equipped with Bluetooth and a sandbox server against the error distance. The error distance is the difference between the actual distance and the distance determined by Bluetooth signals. It helps ensure that the Bluetooth technology can accurately determine the location in a specific environment.

3) Web Page Load Testing: This strategy involves testing the load and performance of individual web pages under various scenarios to ensure smooth flow and an optimal user experience. The experiment simulates multiple users accessing web pages simultaneously and measuring various factors, such as server response time, concurrent user capacity, and page load time. The goal is to ensure that web pages can handle high traffic and load without any failures, because web pages that are slow to load or fail under heavy traffic can lead to a poor user experience and the loss of users or customers.

4) API Load Testing: This strategy involves sending a high volume of requests to the API endpoint concurrently to evaluate the scalability and reliability of APIs used in a web application. In contrast, instead of simulating user interactions with a website, API load testing works by sending requests directly to an API and measuring how it performs. API load testing is important for applications that rely heavily on APIs, such as microservice architectures, mobile applications, and modern web applications. The main objective is to examine its ability to handle concurrent users, analyzing response times and latency under different network conditions that could impact the functionalities and user experience.

Overall, the mentioned strategies are used to assess the performance of the proposed architecture from different perspectives. This can assist in identifying any performance issues for the purpose of overall improvement.

B. Experimental Settings

To show that the proposed architecture is proofed and achieved, all experiments were conducted on a Personal Computer (PC) with an Intel (R) Core (TM) i5-4570 CPU at 3.20 GHz and 8 GB of RAM as a sandbox server placed in an independent and controlled environment. The Xiaomi Redmi Note 11 Pro, based on the Android 11 operating system, was used as a mobile device in all of the experimental testing. Apache JMeter version 5.6.1 was used to simulate various realistic scenarios and implement load testing solutions. Nginx version 1.23.3 was set up as a web server to serve dynamic web pages and web applications written in Python and Java. The backend data was stored in MySQL Server version 8.0.31, which was running on an Ubuntu 22.04 LTS Linux server.

The proposed architecture was evaluated based on the four strategies mentioned above. For the Bluetooth signal testing, a mobile device and a sandbox server equipped with a Bluetooth adapter were used within the indoor office room to record raw RSSI values at varying distances from 1 to 10 meters. The average was calculated within a certain time frame from the raw RSSI values obtained from multiple samplings. The recorded RSSI values were then used to analyze the strength of the Bluetooth signal at different distances. The results were compared to the expected signal strength to determine the reliability of the Bluetooth connection within the office environment. For the location determination testing, a testing scenario was set up where a Bluetooth adapter was placed in a fixed position. The RSSI values were used to estimate the distance between the Bluetooth adapter and a mobile device within the indoor office room to record raw RSSI values at varying distances from 1 to 10 meters. The recorded RSSI values were compared with a pre-built signal strength database to estimate the distance of the mobile device from the Bluetooth adapter. The estimated distances were compared with the actual distances to calculate the error distance for each measurement. For the web page load testing, a testing scenario was set up for evaluating the performance of web pages under the peak load of 500 virtual users. It simulated virtual users performing a complete action on two individual web pages that reflect different types of user interactions, including a landing page labeled as *WebPage1* and a contact page labeled as *WebPage2*, within 60 seconds. The ramp-up period was set to 10 seconds, meaning that the test would start with a few virtual users and increase gradually over 10 seconds until it reached 500 virtual users. In API load testing, a testing scenario was created to identify the maximum load of APIs hosted on a sandbox server and potential bottleneck issues. It simulated 1,000 virtual users to execute different scenarios of APIs, including a topic recommendation API labeled as *API1* and a registration API labeled as *API2*, within 100 seconds. The ramp-up period was set to 5 seconds. In addition, the expected response time was less than 200 milliseconds for 95% of requests.

VI. Experimental Results and Discussion

In this section, the experiments were run on four main strategies to test and evaluate the performance of the proposed architecture. The experimental results demonstrate that the proposed architecture performs well under varying loads and scenarios, meeting the expected response time criteria for the majority of requests. Overall, the experiments validate the effectiveness of the proposed architecture in handling API calls efficiently.

A. The Result of Bluetooth Signal Testing

The experimental results of the proposed architecture in terms of web Bluetooth signal testing were reported to address RQ1. The RSSI measurement is considered to determine the strength of Bluetooth signals. It is a metric that represents the relative quality level of a Bluetooth signal received on a mobile device and a sandbox server equipped with a Bluetooth adapter. RSSI values also influence the range of a reliable Bluetooth connection.

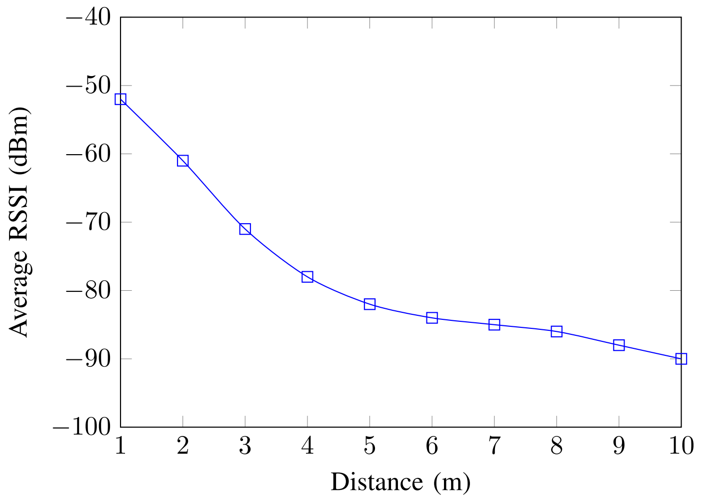

Fig. 9. The relationship between average RSSI and distance

Figure 9 demonstrates the average of the RSSI values corresponding to the different distances ranging from 1 to 10 meters. From the graph, the average RSSI values are around -50 dBm to -90 dBm. The graph shows that the average RSSI values significantly increase when the value of the distance range is changed. Therefore, from this experimental result, the proposed architecture for Bluetooth signals can support Bluetooth connections within 10 meters stably.

B. The Result of Location Determination Testing

The experimental results of the proposed architecture in terms of location determination testing were reported to address RQ2. This testing involves comparing the actual distance between a mobile device and a sandbox server equipped with a Bluetooth adapter against the error distance.

Figure 10 visualizes the results of the Bluetooth location determination test. It compares the estimated distances with the actual distances from 1 to 10 meters and displays the error distances for each point. The blue line represents the estimated distances from the RSSI values compared to the actual distances. This line attempts to closely follow the actual distances, indicating the effort of the sandbox server to accurately estimate the distance between the mobile device and the server. The red dashed line shows the error distances, calculated by subtracting the actual distance from the estimated distance for each point. The estimated distances generally follow the trend of the actual distances, suggesting that RSSI values can be a useful indicator for estimating distance, even with some error. The error distance varies across different points, highlighting the challenges of achieving consistent accuracy. Therefore, based on this experimental result, the proposed architecture can effectively determine the precise location of a mobile device with the accuracy and reliability of Bluetooth signals.

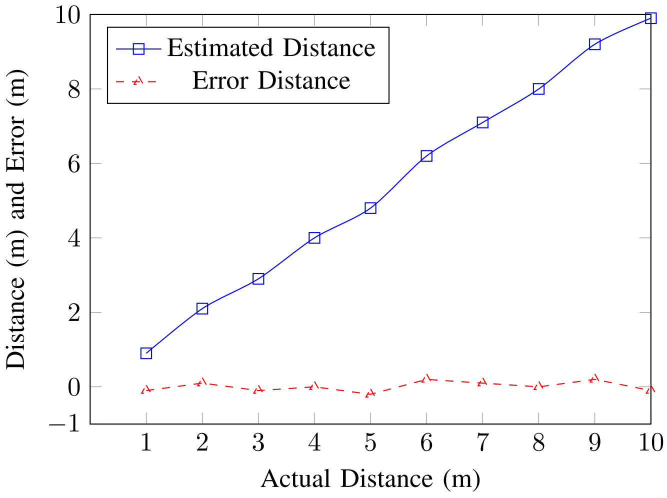

Fig. 10. Bluetooth location determination accuracy test

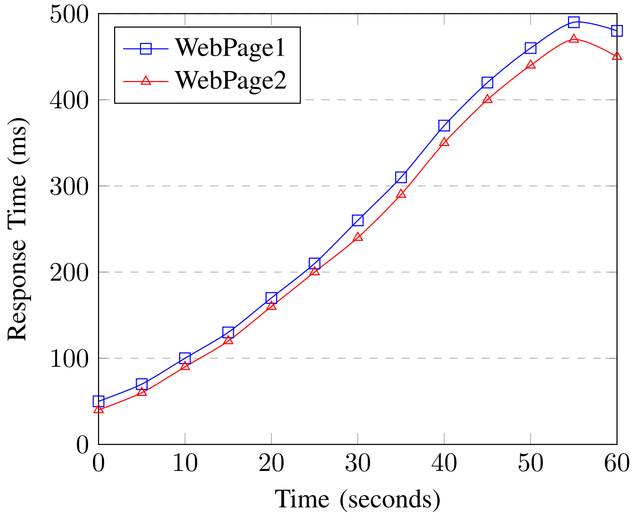

Fig. 11. Web page load testing results

C. The Results of Web Page Load Testing

The experimental results of the proposed architecture in terms of web page load testing were reported to address RQ3 and RQ4. The response time measurement was an important metric to assess the performance of both web pages.

Figure 11 displays the graph representing the result of web page load testing over time. The response time values corresponded to different times ranging from 0 to 60 seconds. The blue line represented *WebPage1*, which started with a lower response time of 50 milliseconds at the beginning, but its response time dramatically increased, reaching a peak of around 490 milliseconds at 55 seconds. After this point, the response time slightly decreased

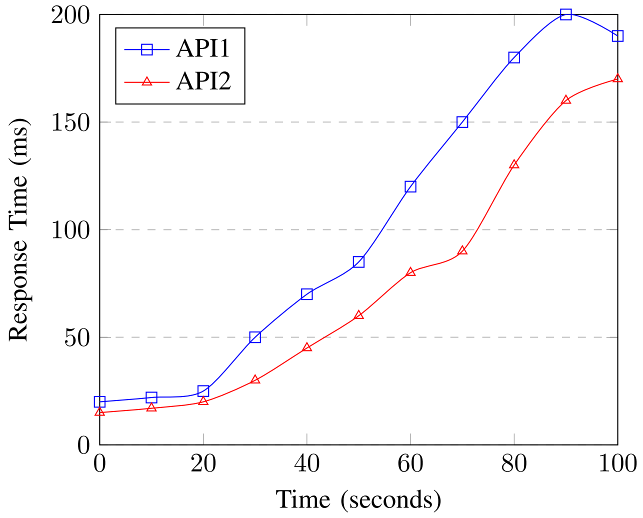

Fig. 12. API load testing results

to nearly 480 milliseconds at 60 seconds. On the other hand, the red line represented *WebPage2*, which had a response time of roughly 40 milliseconds at the start of the test. It reached a peak of approximately 470 milliseconds at 55 seconds. The response time was then reduced slightly to about 450 milliseconds by the end of the test.

In comparison with the two web pages, *WebPage1* always had a slightly longer response time than *WebPage2* because it is the landing page, has a larger data size, and embeds more complex functionalities, images, videos, and audio content, causing it to respond slower under the same load conditions.

This graph indicated that the response times of both web pages increased as the number of concurrent users grew. Another point was that both web pages would be in trouble after about 30 seconds because their response times exceeded the acceptable threshold.

D. The Results of API Load Testing

The experimental results of the proposed architecture in terms of API load testing were reported to address RQ5 and RQ6. The response time measurement was a crucial metric for comparing the performance of the two APIs.

The graph in Figure 12 illustrates the result of API load testing, which was tested over a period of 100 seconds. The response time was in milliseconds, ranging from 0 to 200 milliseconds. Each point on a line represented the response times of APIs at a particular second in the test. The blue line represented the performance of *API1*, whose response time started at 20 milliseconds and gradually increased. There was a significant shift in the response time from 60 seconds to 90 seconds, with a peak at around 200 milliseconds before slightly dropping to 190 milliseconds at the end of the test. The red line represented the performance of *API2*, which started with a low response time of 15 milliseconds. The peak response time was around 170 milliseconds at 100 seconds.

According to these results, *API2* outperformed *API1* under load because *API1* was a recommendation engine, which ran more complicated algorithms and processes than the algorithm of *API2*. From the graph, it could be observed that the response time gradually increased as the number of concurrent users grew. Moreover, both APIs achieved a good response time because it was under 200 milliseconds.

In summary, the proposed architecture achieved remarkable performance for Bluetooth signals, web pages, and APIs. The experimental results show how effectively it works in different situations, indicating that it is good at optimizing performance. It could be deployed in a production environment because it can satisfactorily maintain several factors at acceptable levels, including reliability, scalability, robustness, and efficiency. It can also support many users with large amounts of data. Additionally, the response time of both web pages and APIs remained consistently below the desirable threshold of 200 milliseconds, ensuring a positive user experience. With these scalability strategies in place, the proposed architecture will thrive even under the highest loads that various clients can consume. Furthermore, both web pages and APIs could potentially benefit from performance optimizations, especially in high-load scenarios. This could involve optimizing the backend, such as improving database queries, increasing server resources, using connection pooling to leverage load balancing techniques, or deploying on cloud platforms, and the frontend, such as minimizing JavaScript or optimizing images.

VII. Use Cases

In this section, examples of scenarios from two environments located in a specific area are considered to describe the storytelling and planning processes using the proposed architecture. Integrating mobile computing technology, location-aware technology, and AI techniques in the proposed architecture enhances the user experience by providing personalized services and easy access to resources. These scenarios will demonstrate how users can utilize an application on their smartphones, connected to a sandbox server, to access various services and information tailored to their needs within a controlled environment.

In a conference environment, imagine that hundreds of participants enter a conference room and also open a mobile application in the proximity of a sandbox server. They can make a registration with a sandbox server to be granted resource access through a local Wi-Fi network. The sandbox server identifies the exact location of the participants within the predefined environment and delivers local-based services and information tailored to that location. This means that they will receive necessary services, information, and recommendations generated from the sandbox server in real time, such as a list of participants with profiles, scheduled programs, presentations, documents, videos, and any other relevant material, which will be reflected immediately in the mobile application for participants who are granted access to the network. These services and information are sent proactively and reactively as instant notifications to the participants. Moreover, the mobile application can be customized to allow participants to collaborate with each other who are in the same location environment, meet new friends, make comments, and share information while displaying relevant information, providing a personalized experience for each participant. Therefore, the ability to access information and interact with others through the mobile application can increase user engagement and participation. Finally, when participants stay in the conference environment, they can explore all of the services and information they need. In addition, the full integration between the local sandbox server and the mobile application should be designed to create a comprehensive and engaging experience, making it easy for participants to seamlessly navigate through the conference digitally and identify what topics or items interest them the most. In addition, the local sandbox server can offer personalized recommendations based on preferences, previous conference attendance, and networking profiles of participants, ensuring that they receive relevant and tailored information and make the most of their time in the conference environment. Moreover, future features could include an integration that allows participants to easily access live polls, surveys, and interactive sessions directly from their mobile devices, enhancing engagement and participation during the conference.

Another use case might be in the context of a library. Assume that some students visit a library and have a library application installed on their smartphones. Once the check-in process has been completed, the library application confirms its registration with a local sandbox server. This triggers an automatic connection to a local Wi-Fi network, granting them access to services and resources available while they visit the library. The library application then provides information about available services, such as a dynamic map of the library, book reservations, study rooms, and personalized recommendations based on user preferences and location. For example, if a user moves to the computer science section, the sandbox server recognizes their location and recommends books and resources related to their interests. This includes access to online journals, e-books, and databases hosted on the sandbox server that they can access directly from their device, optimized for fast access over the local Wi-Fi network. Moreover, the user can reserve the availability of nearby study rooms directly from their library application, selecting a time slot and specific equipment they need, like a whiteboard or a multimedia projector. Another innovative feature of the library that students benefit from is the creation of personalized learning pathways. Based on their coursework, research interests, and past interactions with library resources, the sandbox server suggests a tailored set of resources, workshops, and seminars that could enhance their learning and research capabilities.

In summary, the integration of the proposed architecture in the conference and library environments can significantly enhance their conference and library experiences by providing personalized information, services, and recommendations. These case studies illustrate the potential of the proposed architecture for transforming conference and library services, paving the way for future research in these areas.

VIII. Opportunities and Challenges

The integration of intelligent location-aware systems with the existing infrastructure and technologies of urban management systems and business platforms holds great potential for use in various domains within specific areas, such as smart homes, innovative universities, smart cities, smart industries, smart malls, and smart retail. In the context of urban environments, smart cities utilize real-time data and location-based intelligence to create digital landscapes of urban infrastructure, sometimes resulting in a digital twin of a particular city. The proposed architecture is designed as a generic concept that could enable full integration into smart cities, towns, or villages via a mobile application, enhancing citizens' quality of life and making their lives more efficient and convenient. For example, citizens can receive real-time updates on public transportation schedules, traffic conditions, and parking availability, helping them plan their daily journeys more effectively. Additionally, the architecture can offer notifications about upcoming events, cultural activities, and local attractions, encouraging citizens to actively participate in their local community and explore new opportunities for social engagement. In the context of business landscapes, companies can explore deploying intelligent location-aware systems as part of their business intelligence solutions, gaining valuable insights into customer behavior and preferences. This data can be used to tailor personalized experiences and targeted advertising, drive customer satisfaction and loyalty, and ultimately increase revenue. Furthermore, such technology can optimize operations and improve efficiency for businesses and service providers, allowing them to deliver faster and more convenient services to customers.

The main challenges of smart cities include infrastructure and costs. Building and maintaining the necessary infrastructure for smart cities can be complex and expensive. However, the long-term benefits, such as improved sustainability, resource management, and quality of life, make it a valuable investment for both governments and private sector entities. Additionally, collaboration among different stakeholders, including government agencies, technology companies, and community organizations, is crucial to overcome these challenges and ensure the successful implementation of smart cities. In the business landscape, implementing and managing business intelligence solutions can also be complex, requiring significant resources. Companies may need to invest in specialized software, hardware, and skilled staff to effectively collect, analyze, and interpret data. Moreover, ensuring data privacy and security is crucial for protecting sensitive customer information from potential hacking. Nevertheless, overcoming these challenges can lead to a competitive advantage in the market by enabling companies to make data-driven decisions and stay on top of their competitors.

In conclusion, the adoption of intelligent location-aware systems has the potential to transform urban and business environments into more automated, smarter, and more efficient spaces. This transformation can significantly enhance the overall quality and sustainability of life for individuals and the productivity of businesses. However, several challenges in both urban and business landscapes need to be solved in order to fully realize the benefits of intelligent location-aware systems.

IX. Conclusion and Future Directions

The article proposes a conceptual architecture for designing and implementing intelligent location-aware systems for mobile computing environments, focusing on predefined environments, such as shopping malls, universities, coffee shops, or restaurants. The main idea is to recognize and adapt to such an environment and interact with individual mobile users by automatically offering location-based, personalized, and proactive services, information, and recommendations that meet their needs and expectations based on their current location and context when mobile users appear in the vicinity of a sandbox server. The proposed architecture also addresses crucial aspects, including user privacy, data security, and scalability. This means mobile users can store their personal data locally on their own devices, controlling and deciding the location and movement of their data and authorizing service access. Additionally, mobile users can receive potential interests related to an environment through push or pull mechanisms generated by a sandbox server. This approach helps boost productivity and enhance user interaction experiences in specific environments, allowing them to gain deep insights through their mobile devices. According to experimental results, the proposed architecture is smart, privacy-preserving, scalable, and reliable enough for practical deployment in various real-world environments, seamlessly integrating into the daily activities of mobile users.

To improve the performance of location-aware systems and make them smarter, it is important to include and analyze additional contextual data about a location and user, such as time, weather, historical locations, demographics, preferences, interactions, social connections, and behavioral data. This approach enables dynamic adaptations of location-based targeted information and recommendations to the different environments of the user. Furthermore, learning the feedback and behavior of users using reinforcement learning and other techniques is possible to optimize performance and outcomes.

In the future, smart personalized recommender systems should be improved by empowering advanced RAG and LLMs to provide accurate, contextual, and timely answers in human languages to questions for a particular user based on contextual data about the user, such as user profiles, user preferences, user locations, and user interactions. The capabilities of LLMs are that the model can understand the context and intent of a question and a user, which leads to a more accurate, relevant, and satisfying response. For example, a user may receive a promotion message if they have 929,100 miles available. For the flight with plate number TG912 departing from Frankfurt to Bangkok at 7 p.m. on December 30, 2024, an AI model understands that there is an option to upgrade to business class for 30,000 miles or to first class for 90,000 miles. Consequently, the AI model will respond by asking whether the user is interested in this promotion.

Based on this research, future work will involve simulating the proposed architecture in real-world scenarios to examine its performance. This can help identify issues that might need to be apparent in laboratory testing. Real mobile users will also assess the proposed architecture to gather feedback on usability, responsiveness, and the overall user experience. Future research could also explore the integration of emerging technologies, such as IoT, 5G networks, and edge computing, to enhance the capabilities of intelligent location-aware platforms. Another area for future research is the use of advanced machine learning and artificial intelligence techniques to refine the personalization aspects of the system further. This includes adapting to dynamic user preferences and contextual changes in real time.

References

[1] S. Thaiprayoon and H. Unger, “Design of an intelligent location-aware architecture for mobile computing environments,” in *AFIN 2023, The Fifteenth International Conference on Advances in Future Internet*. IARIA, 2023, pp. 11–18. [Online]. Available: https://www.thinkmind.org/index.php ?view=article&articleid=afin_2023_1_40_40015

[2] P. Schneider and F. Xhafa, “Chapter 7 - iot, edge, cloud architecture and communication protocols: Cloud digital ecosystem and protocols,” in *Anomaly Detection and Complex Event Processing over IoT Data Streams*, P. Schneider and F. Xhafa, Eds. Academic Press, 2022, pp. 129–148. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128238189000183

[3] U. Gnewuch, M. Ruoff, C. Peukert, and A. Maedche, “Multiexperience,” *Bus. Inf. Syst. Eng.*, vol. 64, no. 6, pp. 813–823, Dec. 2022.

[4] H. R. Schmidtke, “Location-aware systems or location-based services: a survey with applications to covid-19 contact tracking,” *Journal of reliable intelligent environments*, vol. 6, pp. 1–24, 12 2020.

[5] T. Chen, “Enhancing the performance of a ubiquitous location-aware service system using a fuzzy collaborative problem solving strategy,” *Computers & Industrial Engineering*, vol. 87, pp. 296–307, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835215002193

[6] M. d. C. Rodríguez Hernández, S. Ilarri, R. Trillo, and R. Hermoso, “Location-aware recommendation systems: Where we are and where we recommend to go,” *CEUR Workshop Proceedings*, vol. 1405, pp. 1–8, 01 2015.

[7] S. H. Jang and C. W. Lee, “The impact of location-based service factors on usage intentions for technology acceptance: The moderating effect of innovativeness,” *Sustainability*, vol. 10, no. 6, 2018. [Online]. Available: https://www.mdpi.com/2071-1050/10/6/1876

[8] H. Huang and G. Gartner, “Current trends and challenges in location-based services,” *ISPRS International Journal of Geo-Information*, vol. 7, no. 6, 2018. [Online]. Available: https://www.mdpi.com/2220-9964/7/6/199

[9] Q. Fang, C. Xu, M. S. Hossain, and G. Muhammad, “Stcaplrs: A spatial-temporal context-aware personalized location recommendation system,” *ACM Trans. Intell. Syst. Technol.*, vol. 7, no. 4, pp. 1–30, mar 2016. [Online]. Available: https://doi.org/10.1145/2842631

[10] B. P. Knijnenburg and A. Kobsa, “Making decisions about privacy: Information disclosure in context-aware recommender systems,” *ACM Trans. Interact. Intell. Syst.*, vol. 3, no. 3, oct 2013. [Online]. Available: https://doi.org/10.1145/2499670

[11] S. Renjith, A. Sreekumar, and M. Jathavedan, “An extensive study on the evolution of context-aware personalized travel recommender systems,” *Information Processing & Management*, vol. 57, no. 1, p. 102078, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457319300111

[12] A. A. S. AlQahtani, H. Alamleh, and B. Al Smadi, “Iot devices proximity authentication in ad hoc network environment,” in *2022 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*. IEEE, 2022, pp. 1–5.

[13] L. Li, X. Zhao, and G. Xue, “A proximity authentication system for smartphones,” *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 6, pp. 605–616, 2015.

[14] A. Scannell, A. Varshavsky, A. LaMarca, and E. De Lara, “Proximity-based authentication of mobile devices,” *International Journal of Security and Networks*, vol. 4, no. 1-2, pp. 4–16, 2009.

[15] H.-R. Tsai and T. Chen, “Enhancing the sustainability of a location-aware service through optimization,” *Sustainability*, vol. 6, no. 12, pp. 9441–9455, 2014. [Online]. Available: https://www.mdpi.com/2071-1050/6/12/9441

[16] L. Calderoni, D. Maio, and P. Palmieri, “Location-aware mobile services for a smart city: Design, implementation and deployment,” *Journal of Theoretical and Applied Electronic Commerce Research*, vol. 7, no. 3, pp. 74–87, 2012. [Online]. Available: https://www.mdpi.com/0718-1876/7/3/27

[17] S. Kulkarni and S. F. Rodd, “Context aware recommendation systems: A review of the state of the art techniques,” *Computer Science Review*, vol. 37, p. 100255, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574013719301406

[18] K. Haruna, M. Akmar Ismail, S. Suhendroyono, D. Damiasih, A. C. Pierewan, H. Chiroma, and T. Herawan, “Context-aware recommender system: A review of recent developmental process and future research direction,” *Applied Sciences*, vol. 7, no. 12, 2017. [Online]. Available: https://www.mdpi.com/2076-3417/7/12/1211

[19] X. Meng, Y. Du, Y. Zhang, and X. Han, “A survey of context-aware recommender systems: From an evaluation perspective,” *IEEE Transactions on Knowledge Data Engineering*, vol. 35, no. 07, pp. 6575–6594, jul 2023.

[20] G. Bittencourt, G. Fonseca, Y. Andrade, N. Silva, and L. Rocha, “A survey on review - aware recommendation systems,” in *Proceedings of the 29th Brazilian Symposium on Multimedia and the Web*, ser. WebMedia ’23. New York, NY, USA: Association for Computing Machinery, 2023, p. 198–207. [Online]. Available: https://doi.org/10.1145/3617023.3617050

[21] A. Livne, E. S. Tov, A. Solomon, A. Elyasaf, B. Shapira, and L. Rokach, “Evolving context-aware recommender systems with users in mind,” *Expert Systems with Applications*, vol. 189, p. 116042, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417421013865

[22] M. Casillo, F. Colace, D. Conte, M. Lombardi, D. Santaniello, and C. Valentino, “Context-aware recommender systems and cultural heritage: a survey,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, 08 2021.