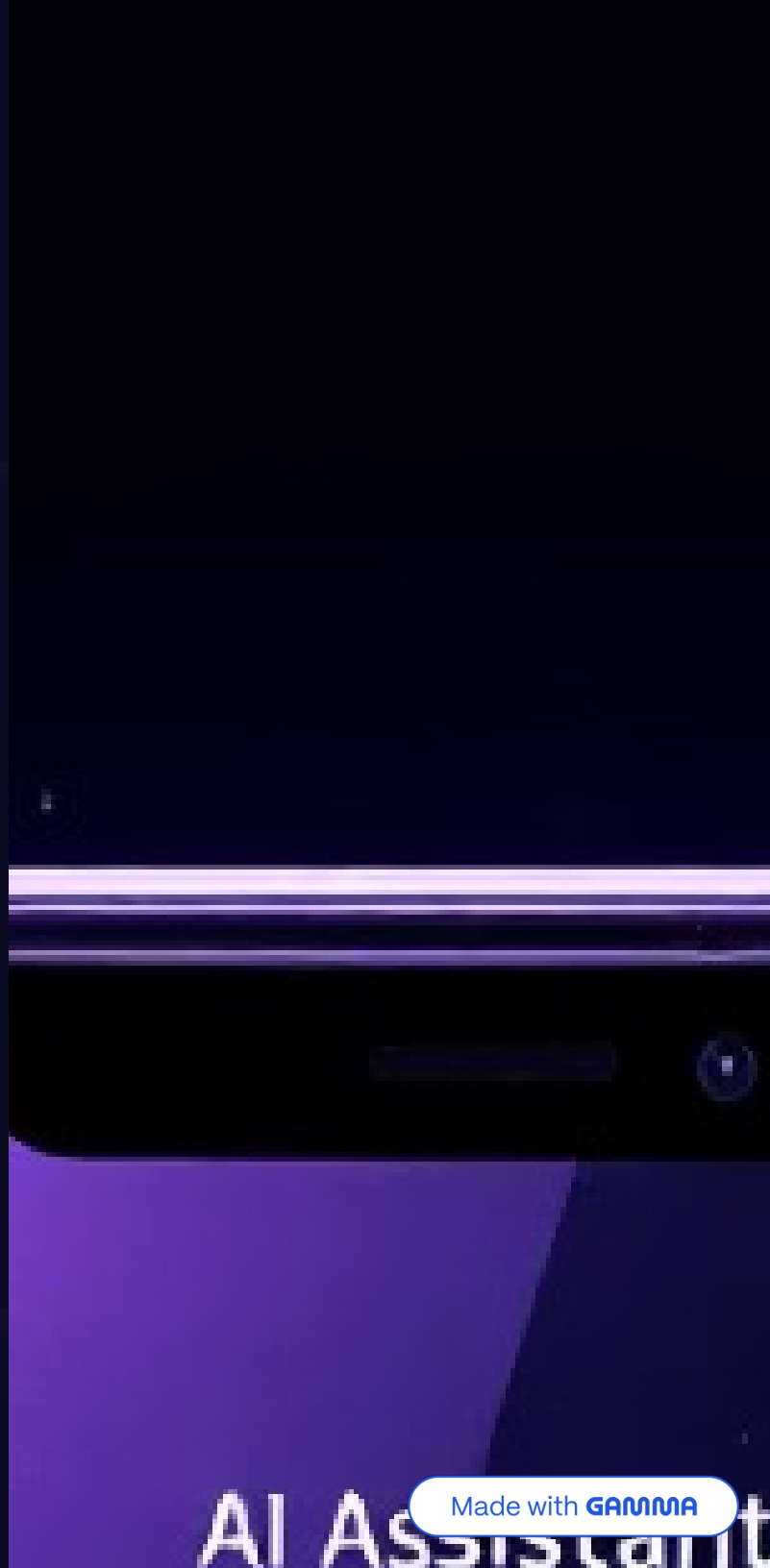
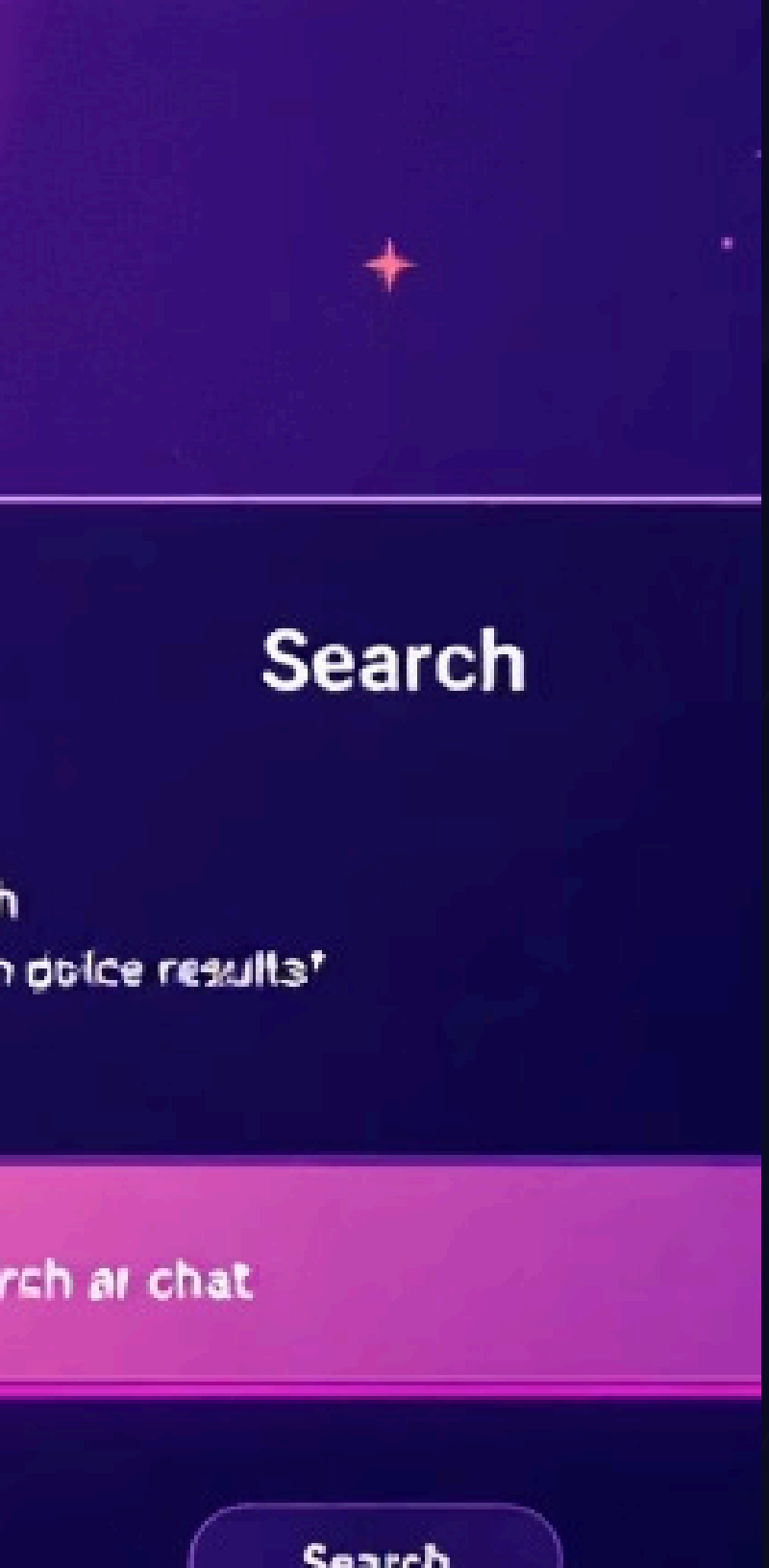
Product Assistant: AI-Powered Shopping Experience

Welcome to Product Assistant, a comprehensive mobile application designed to revolutionize the online shopping experience. Developed using .NET MAUI, this cross-platform solution integrates advanced Artificial Intelligence capabilities to offer a unified and intuitive conversational shopping journey, simplifying product discovery and enhancing user engagement.

1. Product Overview

Product Assistant is built to address the challenges of traditional online shopping by providing a personalized and interactive platform. It acts as a digital shopping companion, leveraging AI to understand user preferences, answer queries about products, and guide users through the purchasing process with natural language interactions.





The Power of Unified Search & Chat

Product Assistant leverages a sophisticated integration of natural language processing and AI-driven search capabilities to provide users with an intuitive and efficient shopping experience. This section details the core functionalities that empower this unified approach.

Natural Language Interaction with AI Assistant

Users can interact with the Product Assistant's AI through natural language queries, mimicking a conversation with a human personal shopper. This eliminates the need for complex menu navigation or precise keyword matching, making the shopping experience more accessible and user-friendly. The AI is designed to understand context, infer intent, and respond appropriately, enhancing engagement.

Intelligent Product Discovery and Grounding Searches

The AI component automatically detects the user's search intent from their natural language input. It then performs "groundi
which involve translating the conversational query into structured

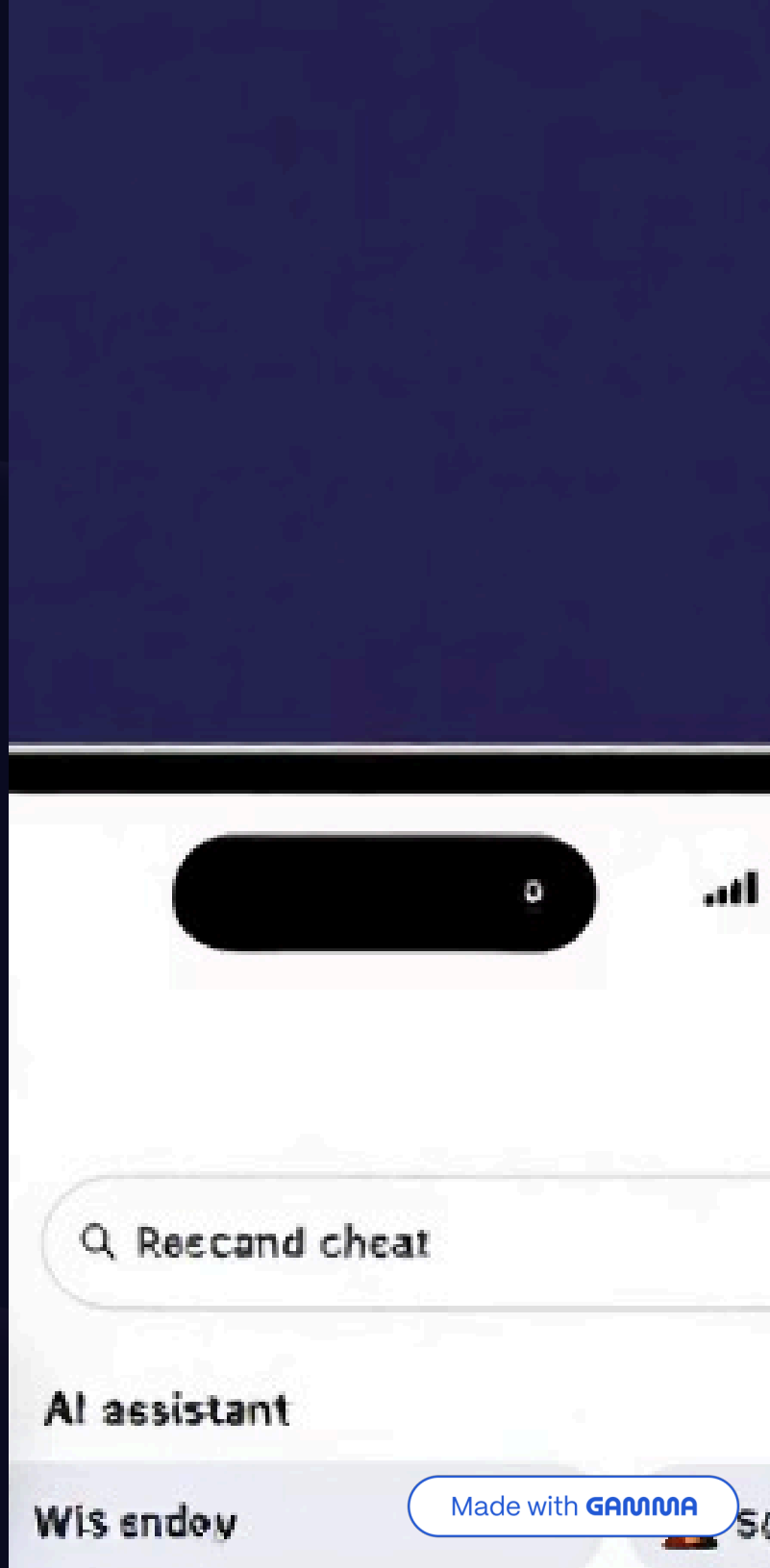
Core Features: Enhancing Your Shopping Journey

This section details the primary features of the platform, designed to streamline and enrich the user's shopping experience through intelligent automation and personalized management.

Unified AI Shopping Assistant

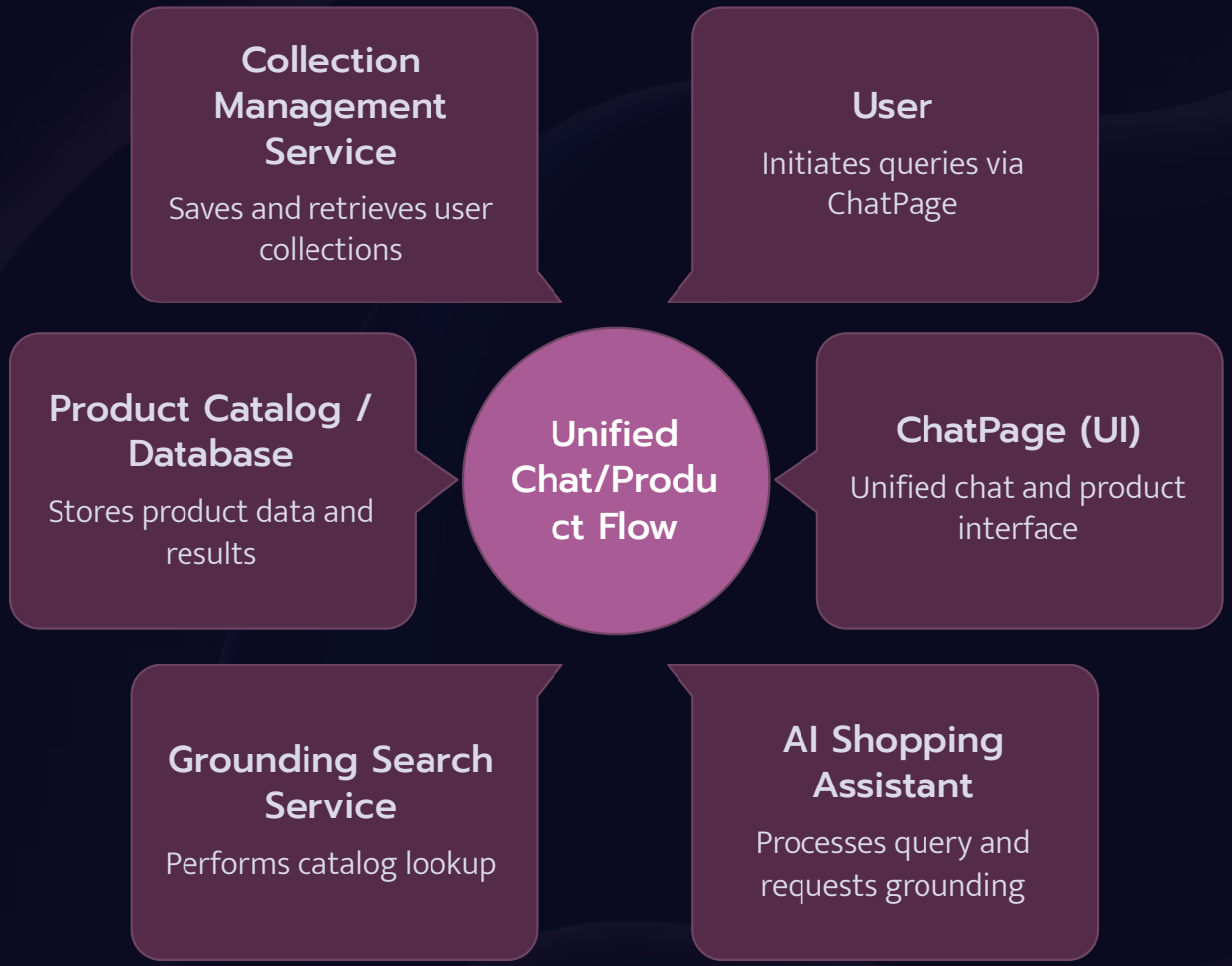
The Unified AI Shopping Assistant provides a cohesive and intelligent interface for all user shopping activities. It integrates advanced AI capabilities to understand user needs, recommend products, and facilitate a smooth purchase journey, all within a natural conversational flow.

- **Natural Language Interaction:** Users can interact with the AI assistant using everyday language, simulating a conversation with a human personal shopper. This eliminates the need to navigate complex menus or use specific keywords, making product discovery intuitive and accessible for all users.
- **Intelligent Product Discovery:** The AI assistant intelligently detects your underlying search intent from your conversations, not just your exact words. It then cross-references your preferences, past data, and product details across a vast catalog to ensure results are highly relevant to your true needs and desires.



System Architecture Overview

This document illustrates how the services communicate and the role of each backend service.

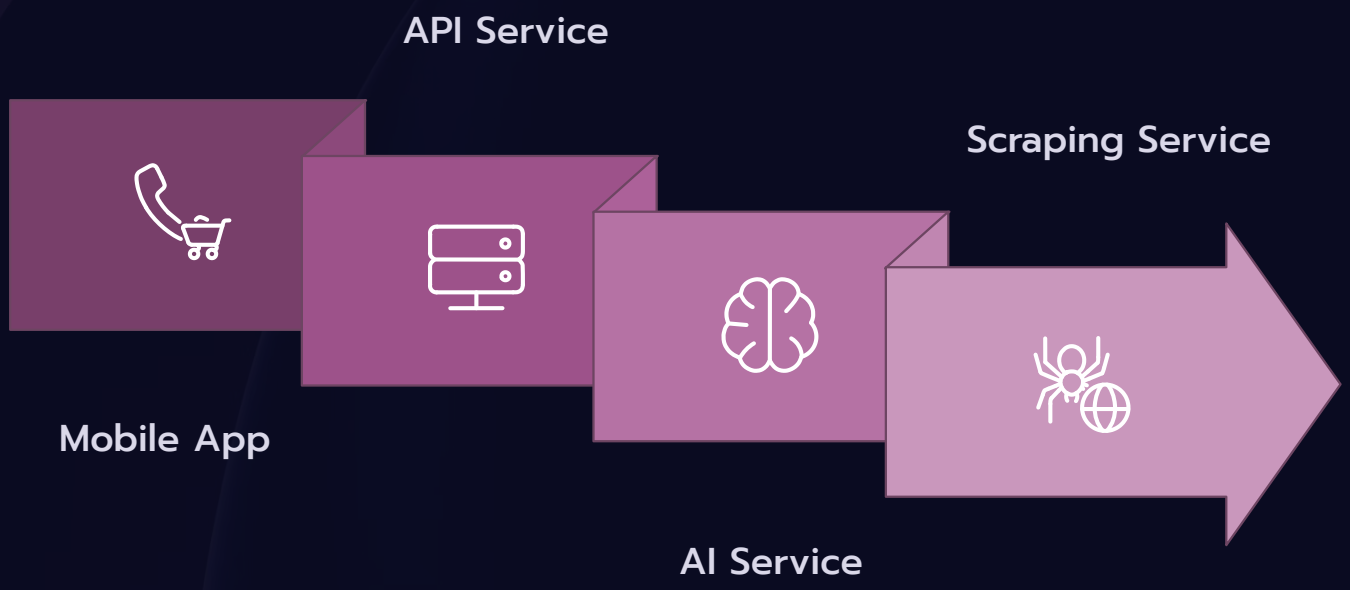


Unified Chat/Product Interface

The application uses a **unified conversational interface** (ChatPage) that combines product search and AI conversation into a

Service Roles & Communication

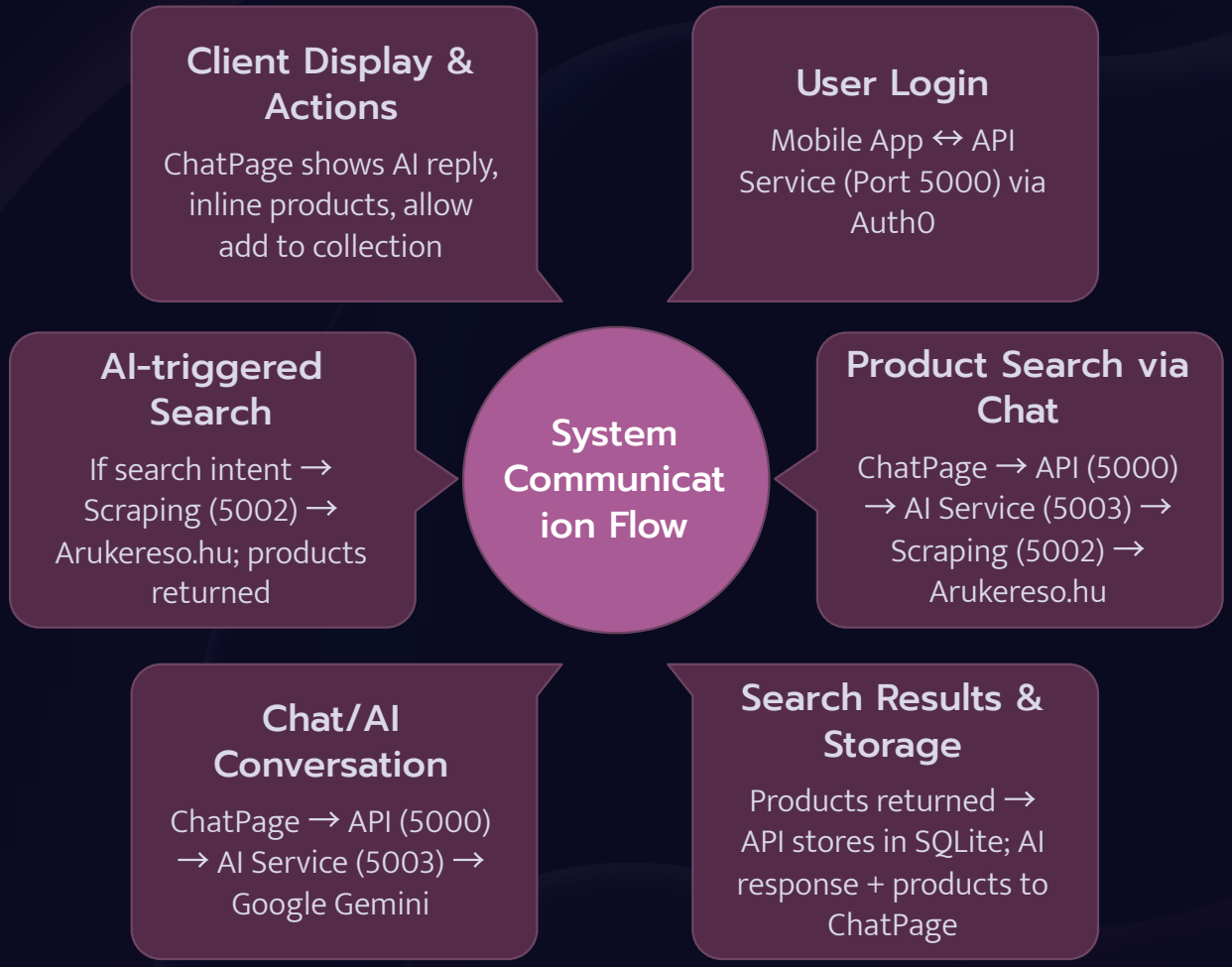
The following diagram illustrates the distinct roles and communication patterns among the core services that power the Unified AI Shopping Assistant. Each service is designed to handle specific functionalities, ensuring a modular and efficient system architecture.



Service Roles

Communication Flow Diagram

This section provides a detailed breakdown of the communication flows within the system, illustrating how various services interact to support key user functionalities such as login, product search, and conversational AI interactions.



Detailed Communication Flows

Request Flow Examples

Example 1: User Searches for "Laptop" via Chat (Unified Interface)

1. User types "Find me a laptop" in the Mobile App chat.

2. The Mobile App sends the chat message to the API Service.

3. The API Service forwards the message to the AI Service for processing.

4. The AI Service processes the message, utilizing the Google Gemini API for understanding.

5. If search intent is detected, the AI Service triggers the Scraping Service to search for "laptop" on Arukereso.hu.

6. The Scraping Service retrieves and parses product information from Arukereso.hu.

Service Responsibilities Matrix & Network Architecture

Service Responsibilities Matrix

Product CRUD	✅ Primary	❌	❌
Product Search (via Chat)	✅ Orchestrates	✅ Detects intent & triggers	✅ Executes
AI Chat	✅ Delegates	✅ Primary	❌
Collection Management	✅ Primary (by UserId)	❌	❌
Authentication	✅ Primary	❌	❌
Data Storage	✅ SQLite	✅ SQLite (Chat)	❌
Web Scraping	❌	❌	✅ Primary
Conversation Memory	✅ Stores	✅ Uses	❌
Grounding Search	✅ Coordinates	✅ Detects & triggers	✅ Executes
External API Calls	❌	✅ Gemini	✅ Arukereso

API Reference Documentation

This section provides a comprehensive reference for all the API endpoints used in the Product Assistant system. It covers authentication, product management, chat and AI interactions, and scraping services, along with details on error responses and rate limiting.

API Service Endpoints

The API Service acts as the primary gateway, orchestrating requests between the mobile application and other backend services.

Authentication Endpoints

POST /api/auth/login

Authenticates a user via Auth0 and returns access tokens.

Request fields:

- `email`: User's email address.
- `password`: User's password.

Successful response includes:

Architecture: Powering the Experience

Our application is built on a robust microservices architecture to ensure scalability, maintainability, and resilience. This modular design allows independent development, deployment, and scaling of individual services. Furthermore, the client-side application adheres to the MVVM (Model-View-ViewModel) pattern, promoting a clean separation of concerns and enhancing testability.

Key Architectural Components

API Service

The API Service acts as the central gateway for all client requests, orchestrating interactions between various backend microservices. It handles request routing, authentication, and response aggregation, providing a unified interface to the mobile application.

Responsibilities:

- Exposing RESTful endpoints for the mobile application.
- Managing user sessions and authentication.
- Calling and coordinating responses from the AI Service and Scraping Service.

Key Technologies Driving Innovation

Our application leverages a carefully selected stack of modern technologies to deliver a robust, scalable, and innovative user experience. This section provides a detailed overview of each core technology and its role in our architecture, focusing on high-level concepts and user-facing benefits.

.NET MAUI 8.0: Cross-Platform Mobile Application Development

.NET MAUI (Multi-platform App UI) serves as the primary framework for building our user-facing mobile application. It enables us to develop a single codebase that compiles into native applications for Android, iOS, Windows, and macOS, ensuring a consistent user experience across different platforms while taking advantage of native performance and UI capabilities.

Key Aspects:

- **Unified Codebase:** Write C# and XAML once to target multiple platforms, reducing development time and effort.
- **Native Performance:** Applications are compiled to native code,

Seamless Experience: AI Tools in Action

Our AI assistant is designed to provide a truly intelligent and efficient shopping experience through the integration of advanced tools. At the heart of this capability is the Grounding Search mechanism, which allows the AI to understand natural language requests and retrieve relevant information from external services.

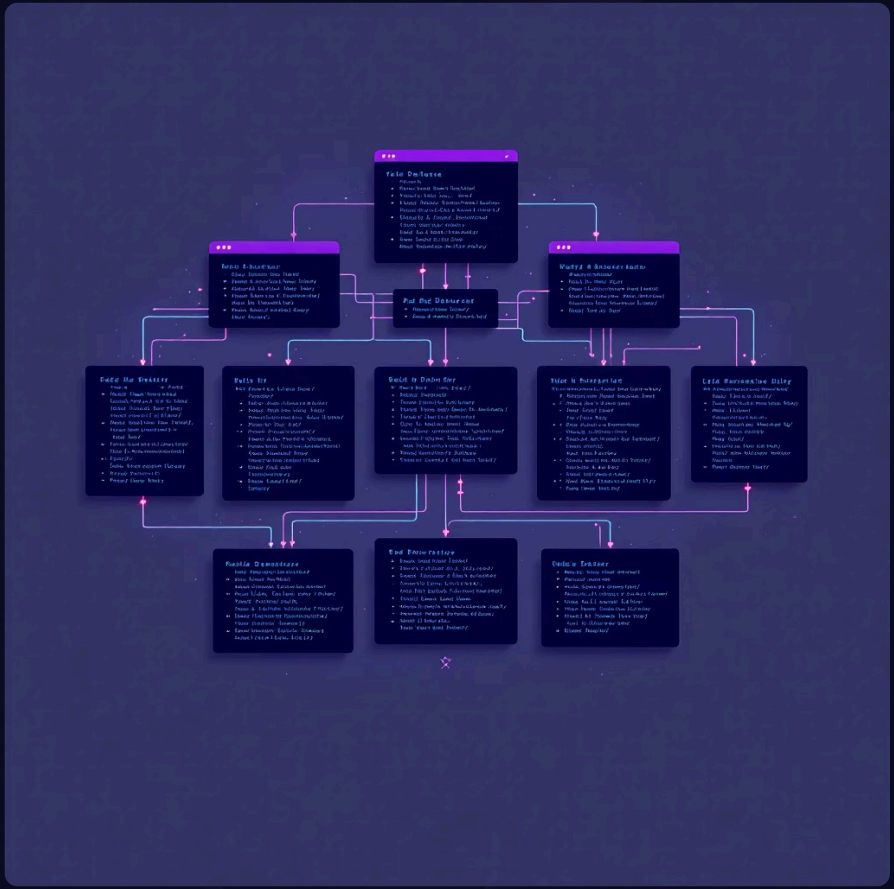
Grounding Search: Bridging AI and Real-World Data

Grounding Search refers to the process where our AI assistant translates a user's natural language query into actionable calls to specialized external tools or APIs. This ensures that the AI doesn't just "talk" about products but can actively "find" and "present" them by interacting with real-time data sources. This capability is crucial for delivering accurate, up-to-date, and contextually relevant results directly within the conversational interface.

The AI Assistant's Workflow: From Query to Interactive Results

The following steps outline the typical interaction flow when a user

Robust Data Model: Organizing Your Information



Our data model is meticulously designed for optimal efficiency, clarity, and security, ensuring that all product and conversation data is managed robustly within the AI assistant ecosystem. It underpins the seamless operation of features like Grounding Search and user-specific product saving. This section details the structure and purpose of our primary data entities and the underlying persistence layer.

Core Entities Overview

The data model primarily revolves around two core entities: the **Product Entity**, which stores details about products found via search, and **Conversation Memory**, which captures the historical context of user interactions. These entities are designed

Device Features: Enhanced Mobile Experience

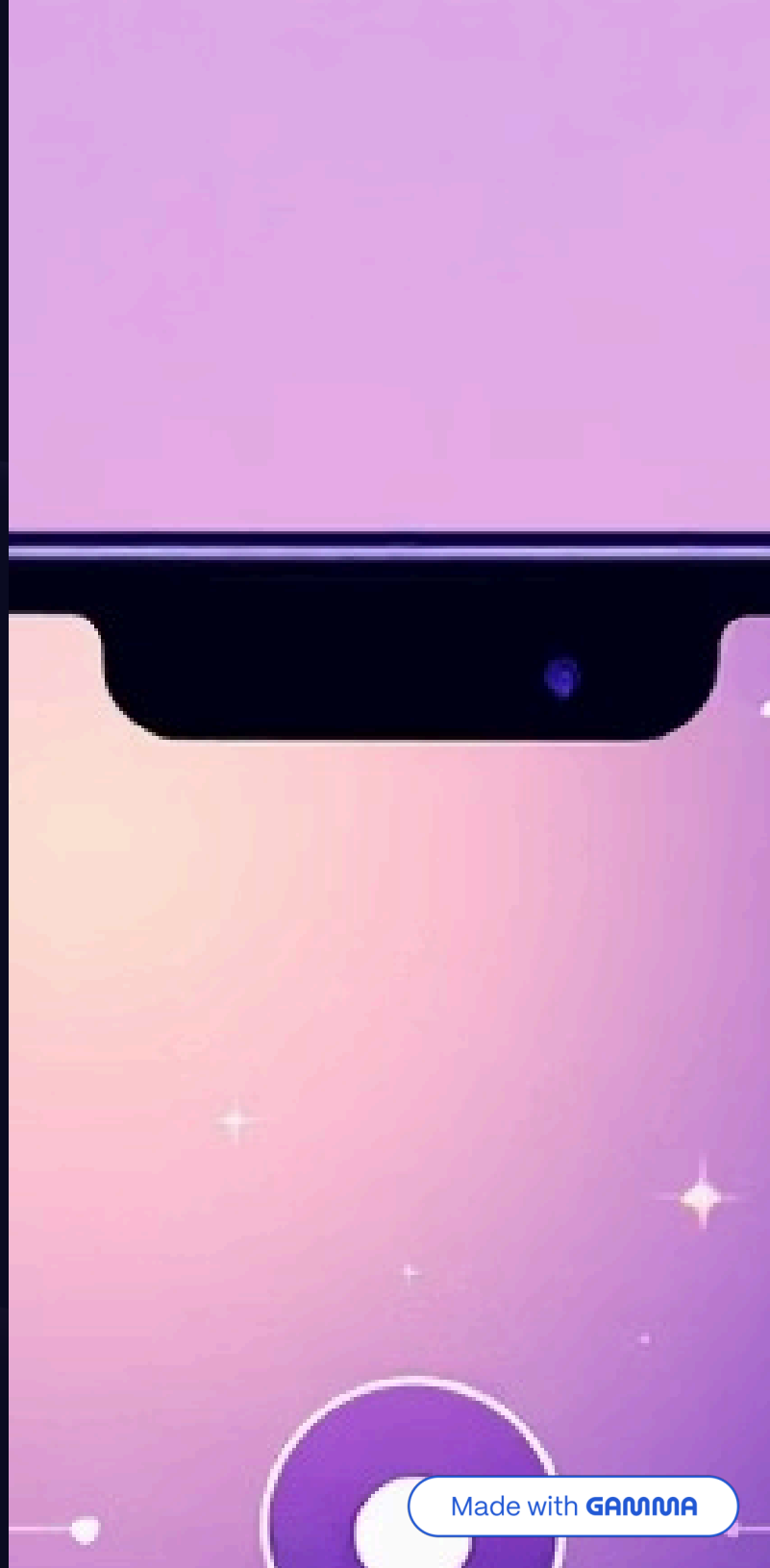
The Product Assistant mobile app integrates essential device features to provide a smooth and contextual user experience. This section details the usage of network connectivity detection and geolocation services within the application, crucial for optimizing performance, conserving resources, and delivering personalized user experiences.

1. Network Connectivity Detection

Effective management of network status is vital for any mobile application that relies on external data. The Product Assistant app continuously monitors the device's network connectivity to ensure optimal performance and resource utilization.

1.1. Real-time Monitoring and Offline Prevention

The application actively monitors network connectivity to respond to changes in real-time. This prevents unnecessary data requests when the device is offline, thereby conserving mobile data, reducing battery consumption, and improving app responsiveness. When connectivity is



Deployment & Scalability for Product Assistant

The Product Assistant architecture is designed for flexible and robust deployment, catering to both local development environments and highly scalable production setups. Leveraging modern containerization and orchestration technologies, it ensures efficient resource management, high availability, and streamlined development workflows.

Local Development with Docker Compose

For local development and rapid prototyping, Product Assistant utilizes Docker Compose. This allows developers to easily spin up all necessary microservices and dependencies with a single command, creating a consistent and isolated development environment. This approach eliminates environment-related discrepancies and accelerates the onboarding process for new team members.

```
docker-compose up -d
```

This command initiates all services defined in the docke compose.yml file, including databases, API services, and any other

Future Enhancements & Support

We are continuously working to improve and expand the capabilities of the Product Assistant, with a clear roadmap for future development and robust support mechanisms in place.

Future Enhancements

Our development roadmap includes several key areas to enhance user experience, performance, and functionality. These enhancements are designed to make Product Assistant an even more powerful and versatile tool for product management.

AI Model Integration & Notifications

- **Enhanced AI Model Selection:** Users will gain the ability to choose from a wider array of AI models for product analysis, allowing for more tailored insights and the ability to switch between models easily.
- **Push Notifications for Price Alerts:** Implement a robust push notification system to alert users to significant price changes or stock availability for tracked products.

Core Feature Expansion & User Experience