



OpenTiny GenUI

Real-time Personalized Generative UI
Reshaping Interaction in AI era

AI 时代的实时个性化生成式 UI 框架，重新定义人机交互体验

Ruihong Lin, Huawei

 OpenTiny

Agenda: Generative UI & OpenTiny Roadmap



01

Intro to GenUI

Evolution of AI
Interaction & Core
Concepts



02

Implementation

Architecture, Core
Components &
Customization



03

Protocol Comparison

What Makes GenUI
Unique from A2UI



04

Future Roadmap

Vision of AI-Native
Generative MiniApps



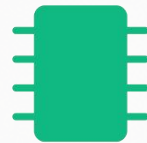
05

Quick Start Demo

Hands-on Guide to
OpenTiny GenUI

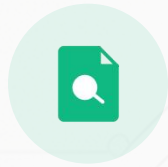
PART 01

Introduction to Generative UI



From Static Layouts to Dynamic, AI-Generated Experiences

The Limitations of Traditional Text-based Chat



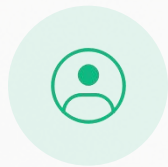
Information Overload

Complex requests result in overwhelming walls of text, difficult to parse and interact with.



Poor Task Efficiency

Multi-step tasks require tedious back-and-forth text exchanges, which is slow and error-prone.



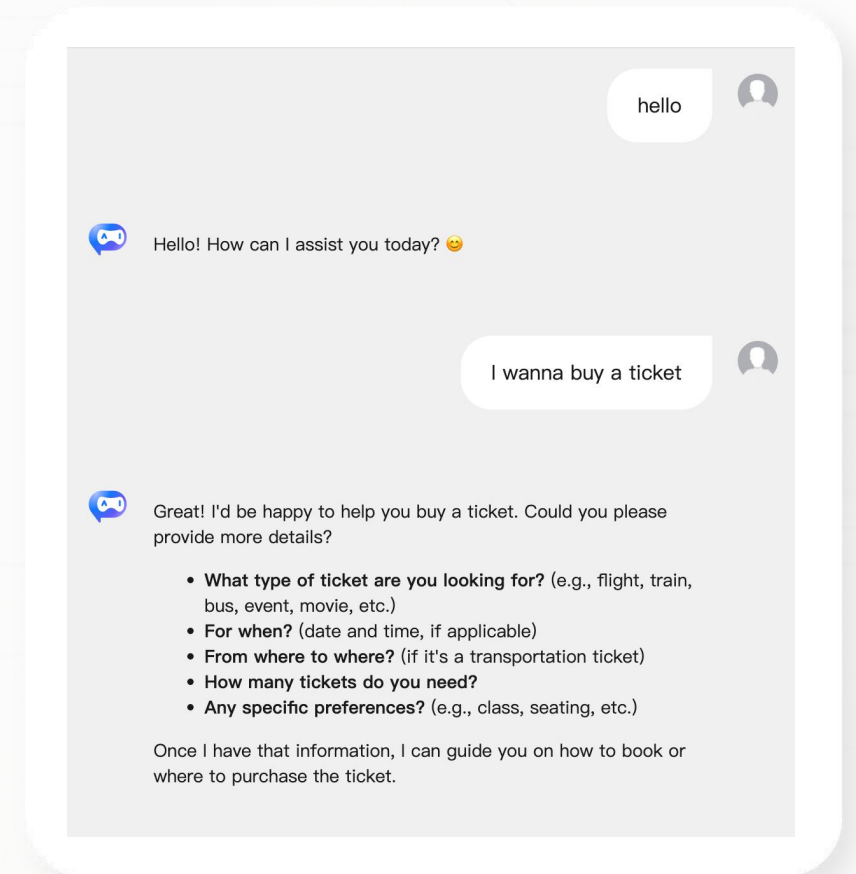
Lack of Personalization

Responses are generic; adapting style or theme doesn't integrate seamlessly.



Business Integration Barrier

Integrating AI into existing processes is clunky, with actions failing to connect directly with backend systems.



Traditional Text Chat Interface

Case Study 1: From Textual Chaos to Visual Clarity



Traditional Text Chat

Static Markdown tables make trend analysis hard. Requires manual copying and reformatting to understand data.



OpenTiny GenUI

Generates interactive **Bar Charts & Tables**. Users sort, filter, and download instantly—turning data into actionable insights.



Sales Data Analysis Report

1. Annual Sales Trend Comparison

Month	2023 Sales	2024 Sales	2025 Sales
Jan	801	1,200	1,308
Feb	665	808	1,101
Mar	1,202	1,446	1,548
Apr	1,056	1,162	802
May	1,466	1,603	1,989
Jun	1,515	1,715	1,600

2. Product Market Share

Product Name	Market Share
Product A	50%
Product B	25%
Product C	10%
Product D	10%
Product E	5%

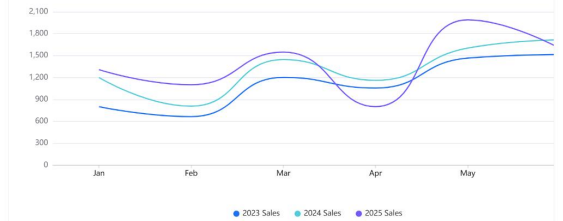
3. Regional Quarterly Sales Comparison

Region	Q1 Sales	Q2 Sales
Southwest	186	721
Central	579	925
North	591	1,356
South	1,006	729
East	517	1,088

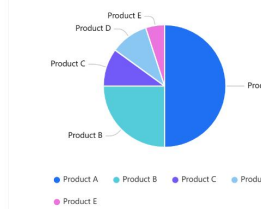


Sales Data Analysis Dashboard

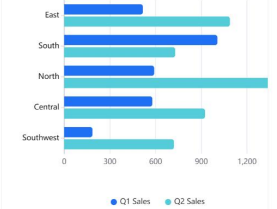
1. Annual Sales Trend Comparison



2. Product Market Share



3. Regional Quarterly Sales Comparison



Interactive Data Dashboard Preview

Transforming Static Data into Intuitive, Actionable Insights

Case Study 2: Optimizing Multi-turn Information Interaction



Traditional Text Chat (Pain Points)

Ordering milk tea requires frustrating text prompts and confirmations. Back-and-forth communication is slow and error-prone.



OpenTiny GenUI (Solution)

Generates **dynamic forms** or interactive cards. Users select directly on the UI, eliminating tedious conversation and mapping choices directly to the system.

Value: Efficiency +60% | Experience Upgrade

I understand you want to order bubble tea. Let me guide you through the options step by step.

Step 1: Tea Type What type of tea would you like? Here are our popular options:

- Classic Milk Tea
- Jasmine Green Tea
- Oolong Milk Tea
- Taro Milk Tea
- Matcha Latte

Please reply with your choice.
(After you select, I'll ask about the next customization option.)

Classic Milk Tea

Step 2: Size What size would you like for your Classic Milk Tea?

- Medium (M)
- Large (L)

Please reply with your choice.

Large

Step 3: Sugar Level How sweet would you like your Large Classic Milk Tea?

- 0% Sugar (No sugar)
- 30% Sugar (Less sweet)
- 50% Sugar (Half sweet)
- 70% Sugar (Regular sweet)
- 100% Sugar (Full sweet)

Please reply with your choice.

30%

Bubble Tea Order

* Tea Type
Classic Milk Tea

* Size
 Medium Large

Sugar Level
30% Sugar

Ice Level
Less Ice

Toppings
 Pearls (Boba) Pudding Grass Jelly Red Bean Aloe Vera

Quantity
1

Special Instructions
e.g., Less sweet, separate toppings...

Submit Order Reset

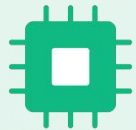
Demo: Smart Bubble Tea Ordering Interface

Case Study 3: Real-time Rendering for Personalized Experiences



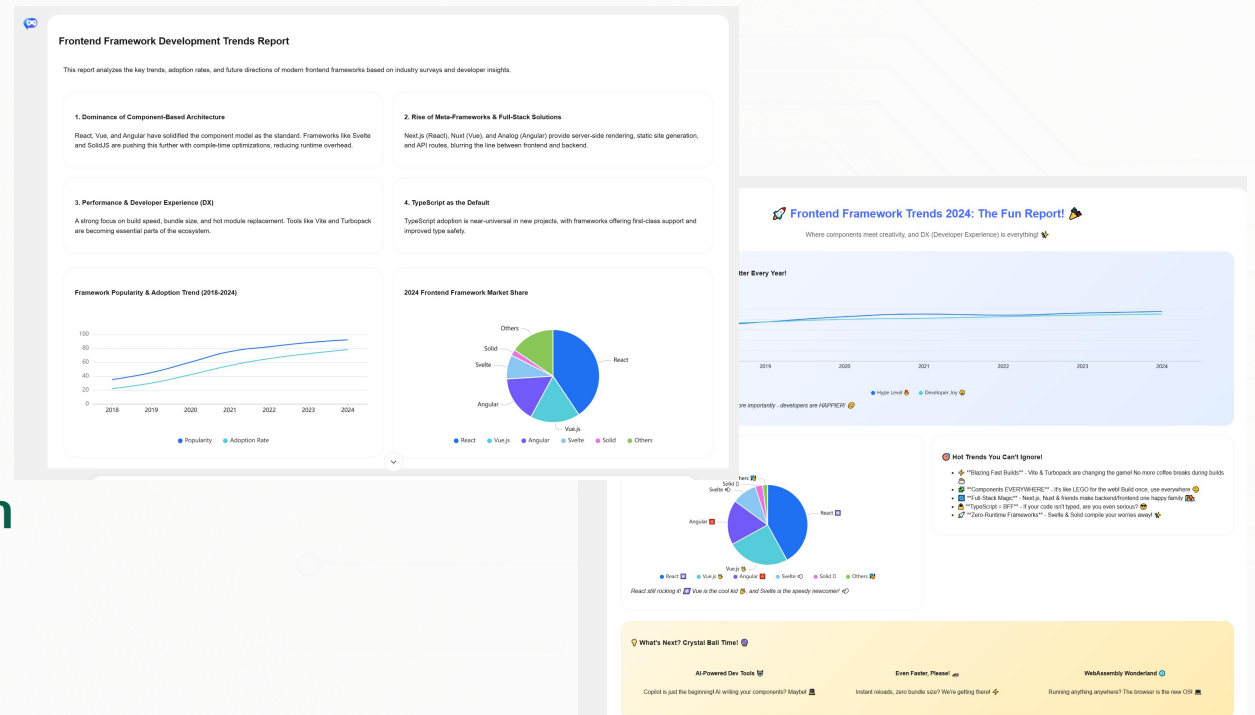
Traditional Text Chat

Style changes only affect prose; visual presentation remains bland and generic.



OpenTiny GenUI: Full UI Generation

- **Formal Style:** Professional fonts, muted colors, structured layout.
- **Lively Style:** Modern fonts, vibrant themes, icons & highlights.



GenUI Generated UI Report Example

Key Insight: Visual personalization is automated without pre-built templates.

Key Advantages of Generative UI

Comparison Aspect

 **Information Density**

 **User Engagement**

 **Task Efficiency**

 **Personalization**

 **Business Integration**

Traditional Text Chat

Low (Linear text stream)

Passive reading experience

Low (Slow, multi-turn process)

Limited to text content/style

Indirect (Needs translation layer)

OpenTiny GenUI

High (Structured & Visual)

Active Interaction (Click/Sort)

High (Direct manipulation)

Deep (Visual theme/layout)

Direct (Maps to business logic)

PART 02

Implementation & Extension Support

A deep dive into the implementation details and powerful extensibility of OpenTiny GenUI, empowering businesses to quickly build robust business foundations.



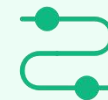
Seamless Integration

Multi-environment
Adaptation



Flexible Extension

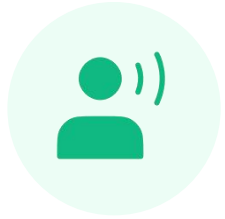
Modular Component
Integration



Context-aware Integration

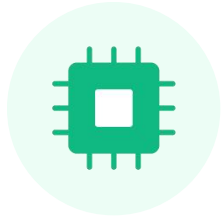
Interaction Context
Extension

Technical Architecture: LLM-Powered UI Generation



User Input

User enters a request in natural language, initiating the generation process.



LLM Core

Processes intent, understands requirements, and outputs structured data (e.g., JSON).



Component Library

A pre-built collection of styled, production-ready UI elements for assembly.



Renderer

Maps the LLM's structured output to library components and assembles the final view.

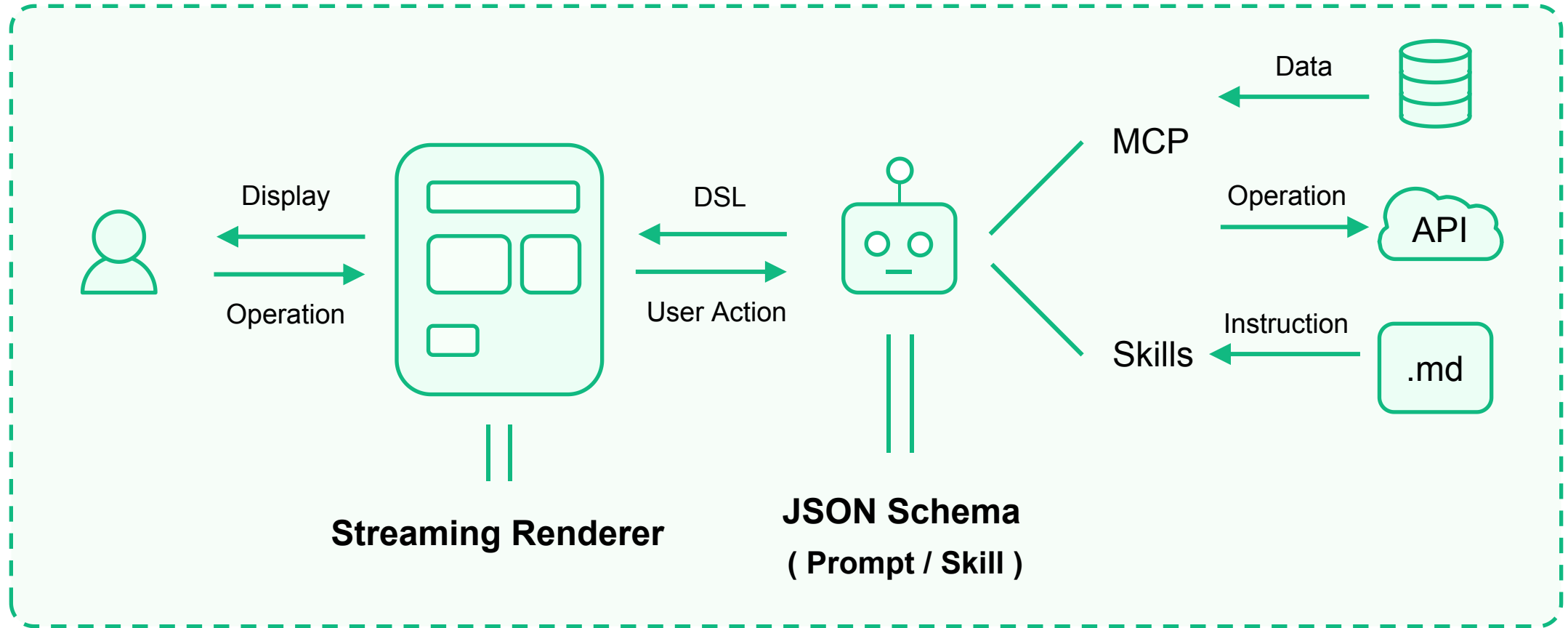


Generated UI

The final interactive interface is displayed, fulfilling the user's request.

Core Insight: Translates natural language intent into structured, interactive UIs seamlessly

OpenTiny GenUI Architecture



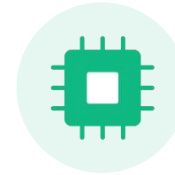
Core Components of GenUI



System Prompt

The "Instruction Manual" for AI

Defines the grammar of UI components for the LLM. It teaches the AI how to interpret requests and format responses as structured UI descriptions.



Streaming Renderer

Real-time UI Construction Engine

Receives AI's structured output stream and parses it to update the DOM instantly. This creates a highly responsive and interactive user experience.

Core Value: Seamless AI-UI Communication & Real-time Responsiveness

Extension Support: Tailoring GenUI to Your Business



Material Customization

Adapt GenUI to your brand & assets

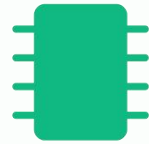
- **Custom UI Components:** Register Vue components like ProductCard or LocationPicker.
- **Brand Style Guide:** Inject your color palette, typography & assets seamlessly.
- **Business Knowledge:** Integrate product catalogs & internal terminology.



Interaction Context

Connect GenUI to your application state

- **Cross-Component Comm:** Trigger actions across different parts of your app.
- **State Management:** Connect with Vuex/Pinia for bidirectional data flow.
- **Context-Aware:** Provide AI with current context for relevant UIs.



PART 03

Comparison with Other Protocols

OpenTiny GenUI vs. Agent-to-UI (A2UI) Protocols

Application Scenario Design

A2UI and OpenTiny GenUI are both JSON protocols based on declarative UI, aiming to enable AI to autonomously generate UI interfaces for user interaction.

A2UI (Agent-to-UI)

Backend-driven: Relies more on backend agents to handle frontend interface refreshes.

Security Restrictions: Does not allow script transmission for security reasons.

Design Philosophy: Focuses on better cross-platform compatibility.

Example: Sorting requires backend agent responses to refresh data sorting.

OpenTiny GenUI

Frontend Interaction: Has complete frontend interaction capabilities and supports frontend script processing.

Design Philosophy: Focuses more on business implementation and user experience.

Example: Sorting can be directly processed by frontend scripts for timely feedback.

Core Insight: A2UI focuses on agent-driven UI display and security, while OpenTiny GenUI focuses on user experience and interaction fluency.

PART 04

Future Roadmap: Generative MiniApp

OpenTiny GenUI serves as the foundation for the next generation of software

What is a Generative MiniApp?

A lightweight, AI-native application dynamically created on-demand to serve a specific user intent.

Example: Family Hiking Trip Planner



Interactive Map: Shows kid-friendly trails & points of interest

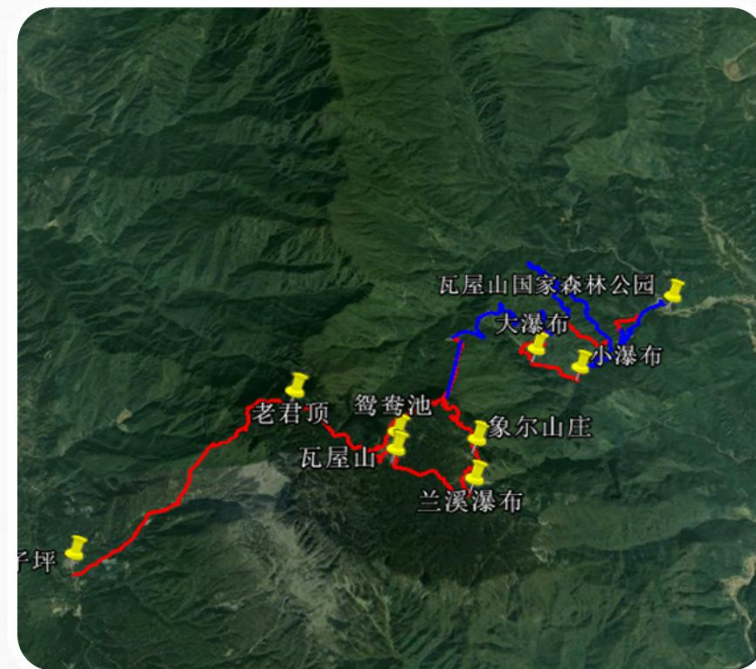


Booking Widget: Real-time cabin/hotel availability & booking



Smart Tools: AI-generated packing list & weather forecast

Key Traits: Ephemeral, Hyper-personalized, Composable & Persistent



AI-Generated Route Visualization

Impact on Software Development



Democratization of App Creation

Reduces the barrier to creating software, allowing non-technical users to "speak" an app into existence.



Agile & Adaptive Software

Applications can evolve and be recreated instantly based on user feedback and changing requirements.



Shift in Developer Role

Developers focus more on creating robust component libraries and defining AI capabilities, rather than building every screen manually.

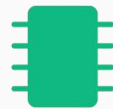
PART 05

Demo: Quick Start with OpenTiny GenUI

Integrate Generative UI into Vue App in Minutes



High Efficiency



Easy Setup



Powerful API

Quick Start: OpenTiny GenUI Vue Integration



1. Initialize Vue Project

```
npm create vue@latest genui-demo && cd genui-demo
```



2. Install GenUI SDK

```
npm install @opentiny/genui-sdk-vue
```



3. Start GenUI Server

```
npm install -g @opentiny/genui-sdk-server  
npx genui-sdk-server
```



4. Modify App.vue

```
Import { GenuiChat } from "@opentiny/genui-sdk-vue"  
<GenuiChat :url="your-api-url" />
```



5. Run the App

```
npm run dev
```



Successful Integration



Ensure your API URL is correctly configured in the GenuiChat props to connect to the AI service successfully. Check console logs for any connection errors.

Join Us: Build the Future of UI

We're looking for passionate developers to contribute to OpenTiny GenUI

Join the OpenTiny Community



Contribute to the next-gen
AI-powered UI framework



Shape the future of
modern software
development workflows



Collaborate with a global
team of passionate tech
experts



GitHub: github.com/opentiny

Thank You

Let's Build the Future Together

 OpenTiny [OpenTiny](#)