

Playground

Connect to my AI assistant →

Available Tools

analyze-repository
Analyzes a GitHub repository and returns a structured summary.

explain-architecture
Explains the internal architecture of a GitHub repository with diagrams.

generate-pitch-deck
Generates a slide presentation for a GitHub repository.

compare-repositories
Compares two GitHub repositories side-by-side based on analysis...

explain-architecture https://github.com/alpic-ai/skybridge

✓ Used tool: explain-architecture

Architecture Overview: Skybridge

Project Summary

Skybridge is a fullstack TypeScript framework for building ChatGPT Apps and Model Context Protocol (MCP) apps with interactive, React-based widgets embedded in AI conversations. It provides end-to-end type inference from the server through to the web client.

Key Architecture Components

1. Skybridge Core Package

- The main published framework package
- Exports both **server** and **web** modules with clean separation
- Provides CLI and build/dev tooling
- Enforces end-to-end type inference

2. CLI & Dev Workflow

- oclif-based CLI** for commands: `dev`, `build`, `start`
- Monorepo scripts for managing workspace
- Template handling and telemetry integration
- Single-command development: `skybridge dev` serves MCP endpoint, Vite HMR frontend, and local emulator

3. Create-Skybridge Scaffolder

- Project generator following `npm create skybridge@latest` pattern
- Pulls templates and prompts users for setup
- Quick-start path for new apps

4. DevTools Package

- Standalone UI and emulator server
- Local widget testing and debugging
- Integrates with MCP SDK and workspace package

5. Example Applications

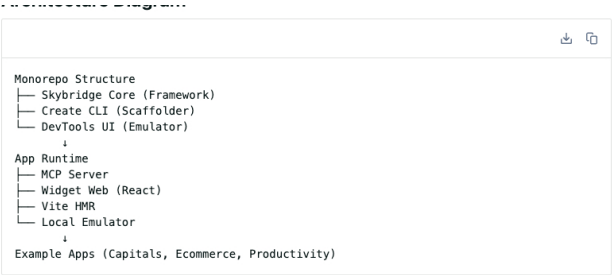
Multiple reference implementations:

- Capitals** - Basic widget showcase
- E-commerce Carousel** - UI composition patterns
- Everything** - Comprehensive feature demo
- Manifest UI** - Advanced configurations
- Productivity** - Real-world use case

Technology Stack

Layer	Technology
Frontend	React + React Hooks (useToolInfo, useCallTool)
Backend	Express.js
Build/Dev	Vite with HMR
Protocol	MCP (Model Context Protocol) SDK
Validation	Zod for schema validation
State Management	Zustand
CLI Framework	oclif
Analytics	PostHog (telemetry)
Language	TypeScript with TSX

Architecture Diagram



Key Patterns

- 1. **Monorepo Workspace** - Multiple publishable packages
- 2. **Fullstack Shared-Types** - End-to-end type safety
- 3. **Hook-based UI Composition** - React-first widget building
- 4. **Template Scaffolding** - Quick project generation
- 5. **Command Pattern** - CLI-driven workflows

Interesting Design Decisions

- **Dual Entrypoints:** `./server` and `./web` modules enforce clean separation of concerns while maintaining type inference
- **One-Command Dev Loop:** `skybridge dev` simultaneously serves the MCP endpoint, frontend HMR, and local emulator
- **Large Monorepo:** 386+ files organized efficiently with separate build pipelines for core, scaffolder, and devtools
- **Skill Evals:** Includes ChatGPT app builder skill evaluation artifacts for AI-driven development

This architecture is optimized for rapid iteration on AI-embedded interactive applications with strong type safety and excellent developer experience.



Send a message...

+

↑