Experience Engineering

Design system context + code context = higher-quality output.

Self-hosted Penpot ↔ MCP ↔ Claude Code in VS Code

PERSON
Designer
works in Penpot UI

Edits brand colors, spacing, button radius. Creates new components, marks variants.

edits

PERSON
Engineer
works in VS Code

Asks Claude to build features. Claude sees code + design system simultaneously.

prompts

BROWSER
Penpot Web App
self-hosted

design.acme.io, loads boards, tokens, components. Live edits sync to server.

syncs

IDE
VS Code + Claude Code
extension running locally

Opens the SvelteKit repo. Claude reads code, runs commands, calls MCP tools.

runs

BACKEND
Penpot Server
self-hosted · docker · postgres

Source of truth for files, components, design tokens, library entries.

API

BRIDGE
Penpot MCP Plugin
connects design ↔ Claude

Exposes execute_code, export_shape, penpot_api_info. Claude calls these to read & modify designs.

tools

AGENT
Claude Code
Opus 4.7

Reads code + design context. Generates components, runs tests, edits Penpot, all in one loop.

CONTEXT
Design context
live, queryable

Token values, color scales, components, semantic aliases, read on demand via MCP.

CONTEXT
Code context
SvelteKit repo

tokens-source/*.json, src/lib/components/, src/routes/. Claude reads, edits, runs build:tokens.

OUTCOME
Higher-quality output
shipped faster

Designer edits = code updates. Claude never invents tokens, uses the real ones.

FLOW

1. Designer edits tokens in Penpot UI → Penpot Server
2. Engineer asks Claude in VS Code → Claude reads code AND queries Penpot via MCP
3. Claude exports tokens.json from Penpot → drops into tokens-source/ → npm run build:tokens
4. CSS variables regenerate → Svelte components pick up changes automatically. No manual sync
5. Claude can also create new Penpot components matching new Svelte ones, symmetry in both directions

Roshan Ravi

roshan@roshn.me roshn.me