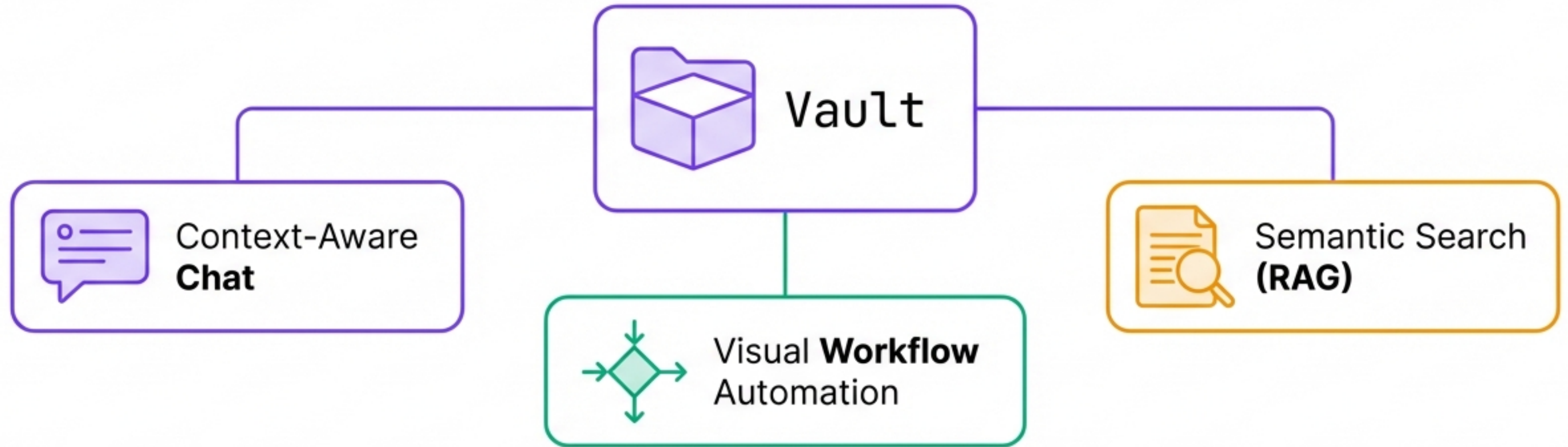


Obsidian Gemini Helper

Comprehensive User Guide & Technical Overview



Core Capabilities



AI Chat

Interactive conversation interface integrated directly into the sidebar.

- Streamed responses with context-aware @mentions
- Multi-modal support: Analyze Images, PDFs, and Text
- Slash commands (`/`) for reusable prompt templates



Visual Workflow Builder

Automate complex multi-step tasks without writing code.

- Node-based visual editor (Drag & Drop)
- 22+ Node Types: Logic, IO, LLM, HTTP
- Natural language generation: 'Create a workflow that...'



Vault Integration

Deep integration with local files and semantic search.

- **RAG**: Semantic search using Google File Search API
- Safety: Edit history with Diff View (Apply/Discard)
- Tools: Create, read, and batch-edit notes securely

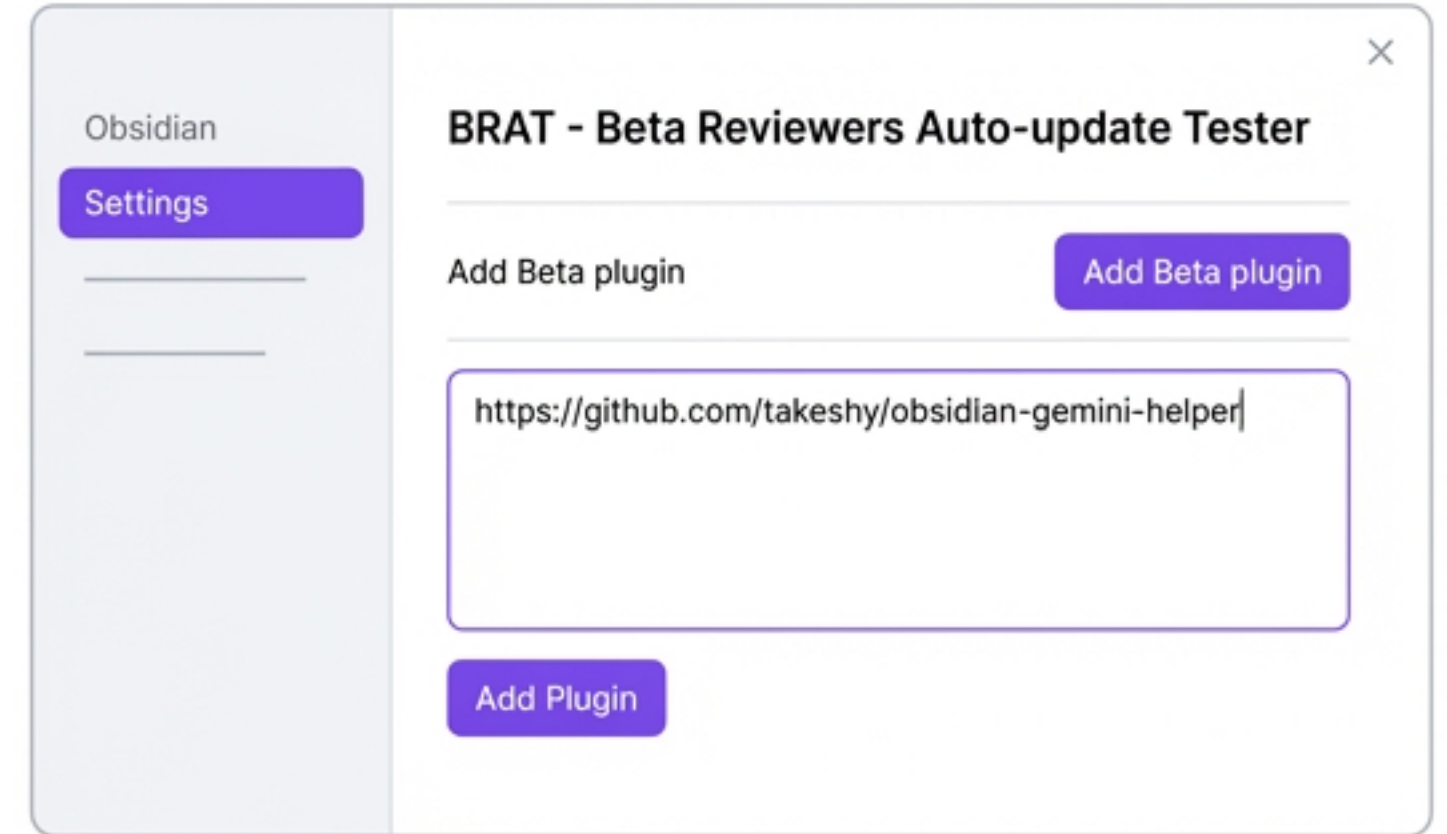
Installation Methods

Method A: Via BRAT (Recommended)

1. Install **BRAT** plugin from Community Plugins.
2. Open BRAT Settings and click **Add Beta plugin**.
3. Paste Repository URL:

```
https://github.com/takeshy/obsidian-gemini-helper
```

4. Enable **Gemini Helper** in Community Plugins.

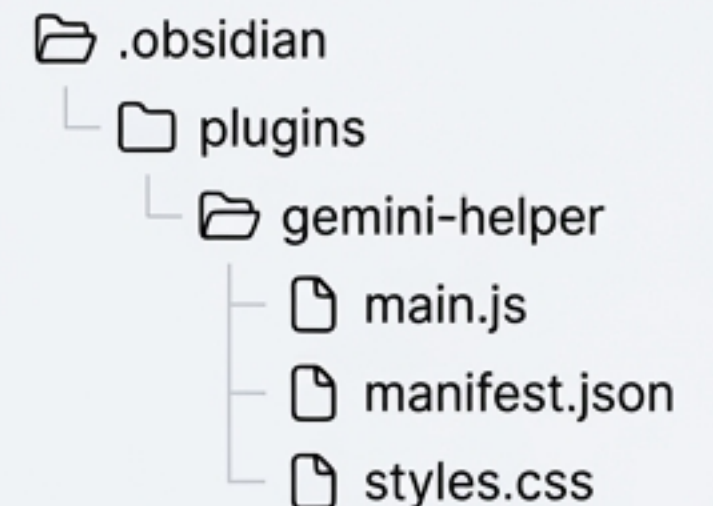


Method B: Manual Installation

Download release files (main.js, manifest.json, styles.css) from GitHub Releases.



Place in .obsidian/plugins/gemini-helper.



API & Model Configuration

Option A: Google API 
(Easiest Setup)

Option B: CLI Mode
(Advanced / Private)

Source

Get Key from ai.google.dev

Cost Model

Free Tier (Rate limited, data used for tuning) OR Paid Tier (Private)

Capabilities

Full RAG Support, Image Generation, Vault Operations

Models

Gemini 1.5 Flash, Gemini 1.5 Pro, Gemini 2.5

Mechanism

Spawns local child processes. Bypasses direct API keys in Obsidian.

Supported CLI Tools

- Gemini CLI: Authenticates via local Google session.
- Claude Code: `npm install -g @anthropic-ai/claude-code`
- Codex CLI: `npm install -g @openai/codex`

Limitations

RAG and Web Search generally disabled. Local context only.

Configure in Settings -> Gemini Helper -> API Settings. Select 'Model' after authentication.

The Chat Interface

Access

Click Ribbon Icon or `Cmd+P` > `Gemini Helper: Open chat`

Context Control (@ Mentions)

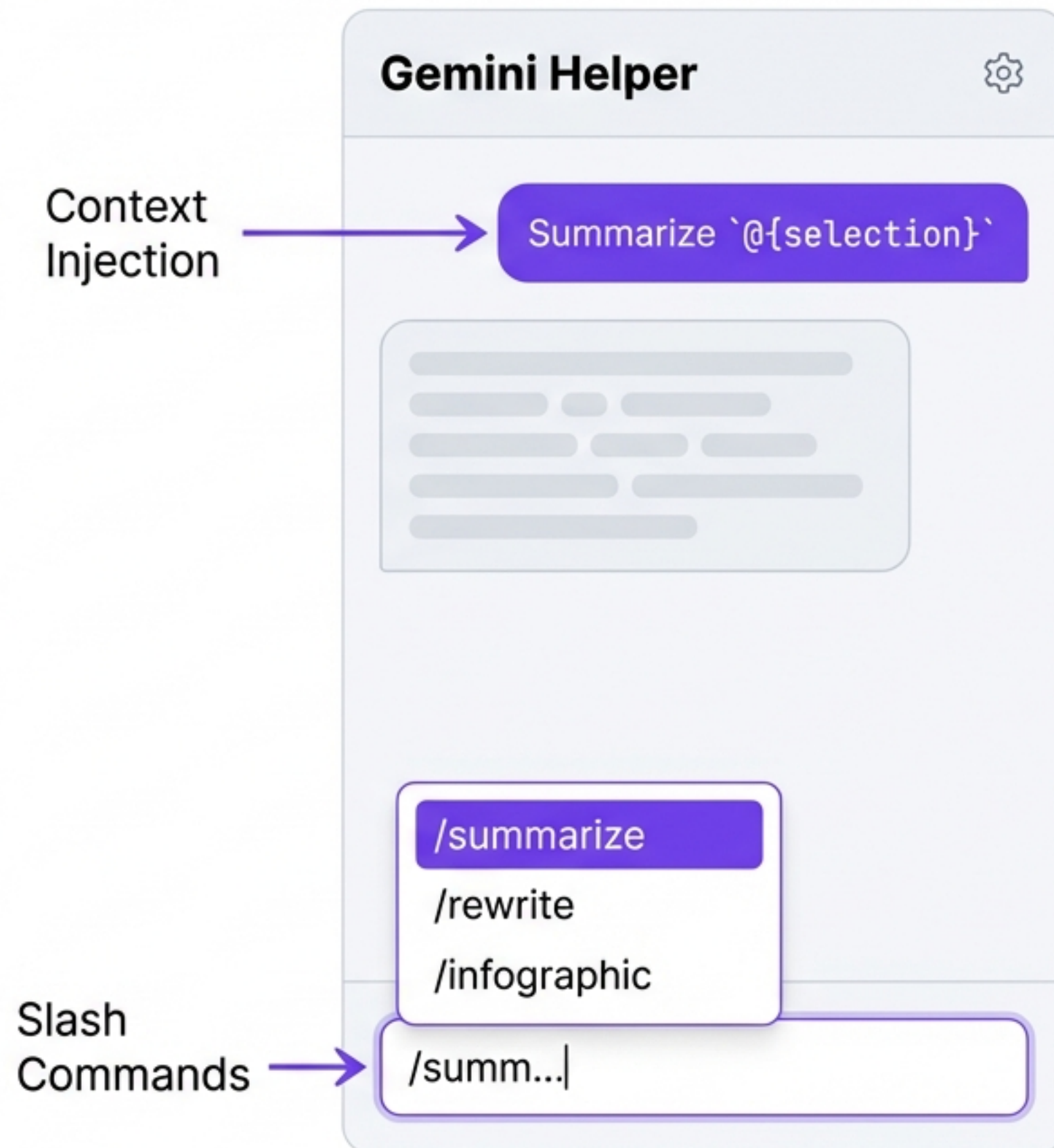
- `@filename`: Reference specific note path. AI reads content via tools.
- `@{selection}`: Pass currently highlighted text.
- `@{content}`: Pass full active note content.

Slash Commands (/)

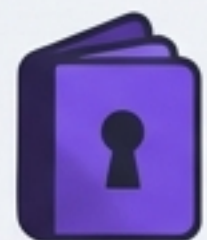
- Use templates for reusable prompts.
- Default: `/infographic` (Converts note to HTML).
- Custom: Define in settings (e.g., `/summarize`, `/rewrite`).

Attachments

Drag & drop images/PDFs directly into chat.



Semantic Search (RAG)



Obsidian Vault

Sync (Incremental)



Google File Search API

(Vector Index)

Concept & Setup

- **What it is:** Uses Google File Search API to index your vault. Allows AI to answer questions based on specific notes.

Modes:

- **Internal Mode:** Plugin manages sync to Google Vector Store.
- **External Mode:** Connect to existing Google Vector Store ID.

Sync & Privacy

- **Sync Behavior:** Incremental uploads (only changed files). Saves bandwidth.

Filtering:

Use Regex to exclude sensitive paths.

Example: `Private/*` or `Secrets/keys.md`

Constraint:

Requires Google API Key. Not available in standard CLI mode.

Safe Vault Editing & History

Diff View

Proposed Changes: project_notes.md

11

12

13

14

15 - Original text line here.

16 + Revised text line proposed by AI.

17

18

19

20

Discard

Apply

Propose & Confirm Workflow

1. AI suggests edit via ``propose_edit`` tool.
2. Diff View appears for review.
3. User explicitly clicks **Apply** to commit.

****Edit History****

- Snapshots saved to ``obsidian/history`` before any AI modification.
- Command: ``Show edit history`` to revert changes.

****Security****

Supports End-to-End Encryption (E2EE) for logs (v1.4+).

Visual Workflow Builder

Concept

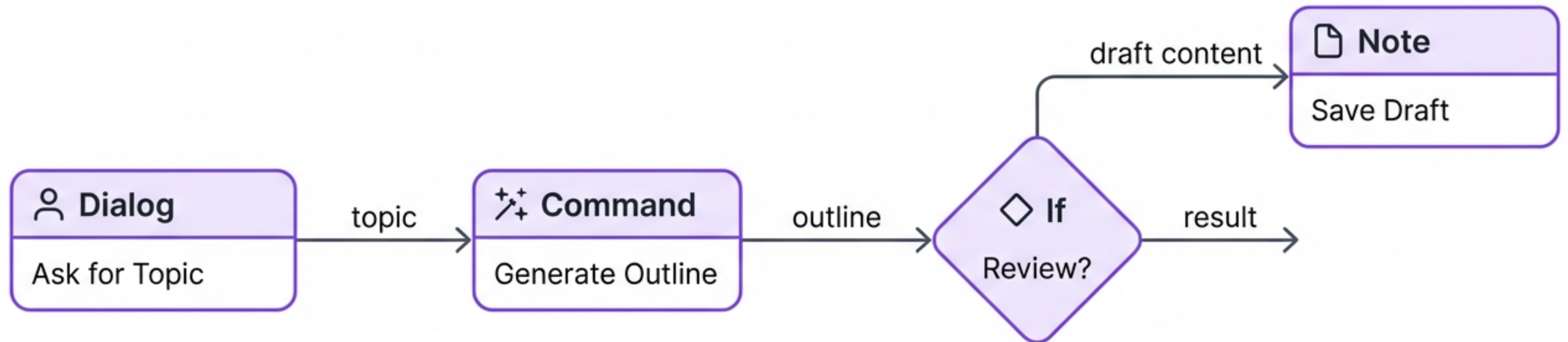
Chain LLM prompts, logic, and file operations into automated sequences. No coding required.

Interface

Accessed via 'Workflow' tab.
Features drag-and-drop visual node graph.

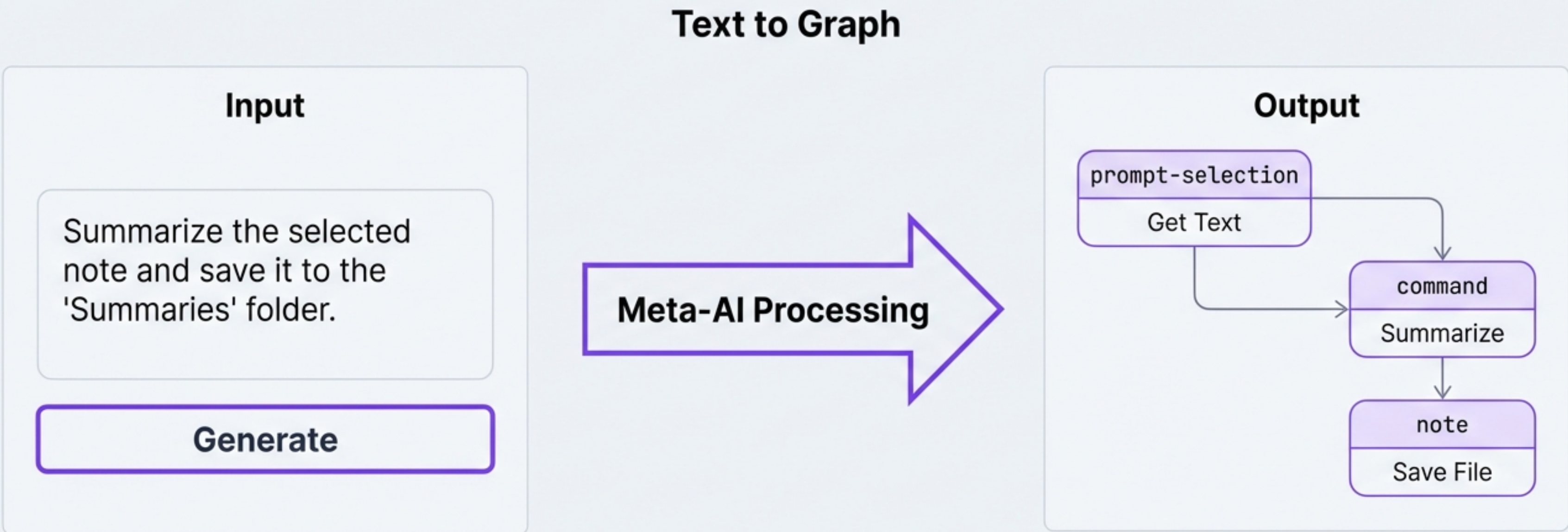
Storage

Workflows are standard
Markdown files using ```
```workflow``` code blocks.





# AI-Powered Workflow Generation

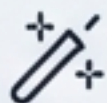


## **\*\*No-Code Creation\*\***

Click **+ New (AI)**, type intent, and let AI construct the graph.

## **\*\*AI Modify\*\***

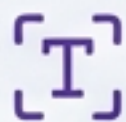
Use the **Sparkle Icon** on existing workflows to request changes (e.g., 'Add a translation step').



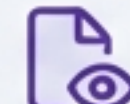


# Essential Node Types Reference

## LLM & Prompts

-  `command`: Execute AI prompt (supports model overrides).
-  `dialog`: Request user input (text/options).
-  `prompt-selection`: Get currently highlighted text.

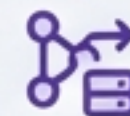
## File Operations

-  `note`: Create or overwrite a Markdown note.
-  `note-read`: Read file content into a variable.
-  `file-save`: Save binary data (images/PDFs).

## Logic & Control

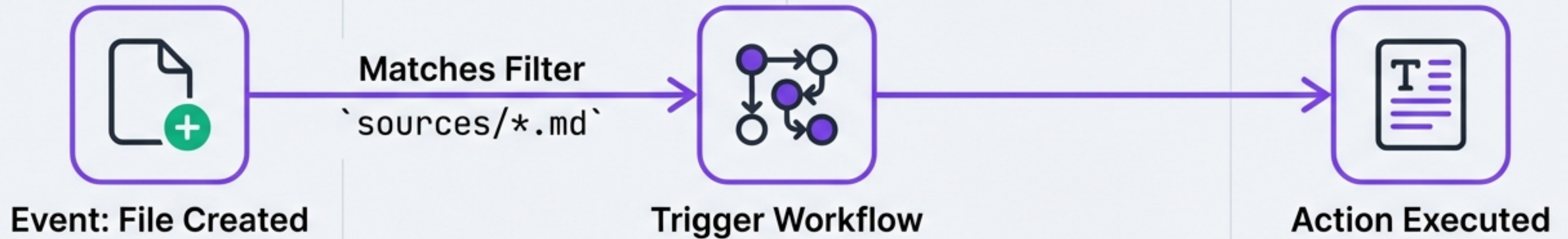
-  `if`: Conditional branching.
-  `while`: Looping structures.
-  `workflow`: Run another workflow (Sub-routine).

## Data & External

-  `http`: Make generic API requests.
-  `json`: Parse JSON data.
-  `mcp`: Call Model Context Protocol servers.



# Event Triggers (Automation)



**Mechanism:** Run workflows automatically on Obsidian events.

**Trigger Types:** File Created, Modified (Debounced 5s), Renamed, Deleted.

**Context Variables:**

\_\_eventFilePath\_\_: Path of triggered file.

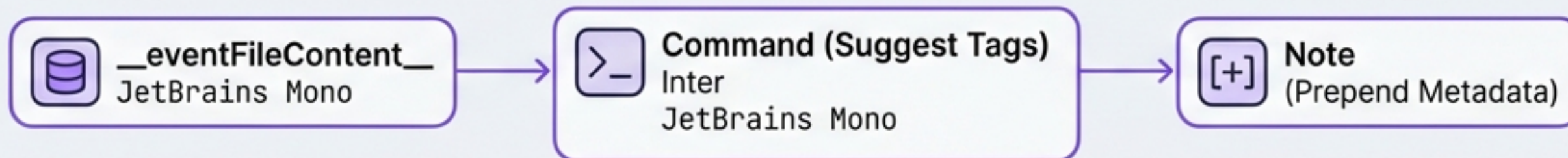
\_\_eventFileContent\_\_: Content of file.

**Setup:** Open Workflow > Click **Lightning Bolt** (⚡) > Define Path Pattern.

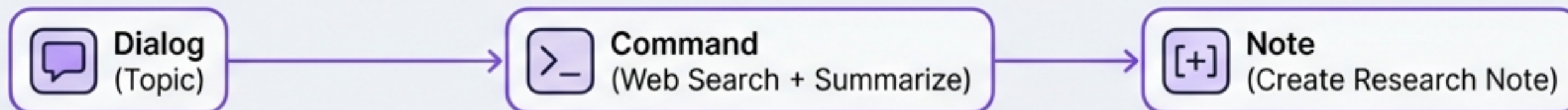


# Practical Workflow Examples

Example 1:  
**Auto-Tagging**  
(Event: File Created)



Example 2:  
**Web Research Assistant**



Example 3:  
**Batch Summarization**

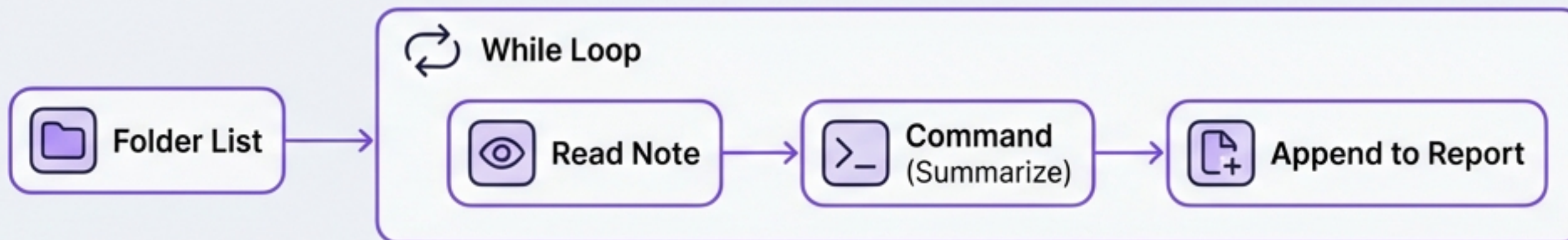
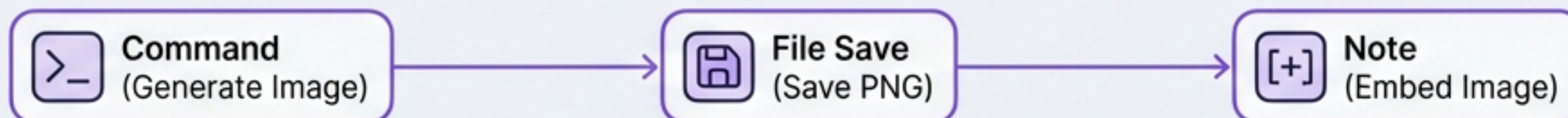


Image Generation



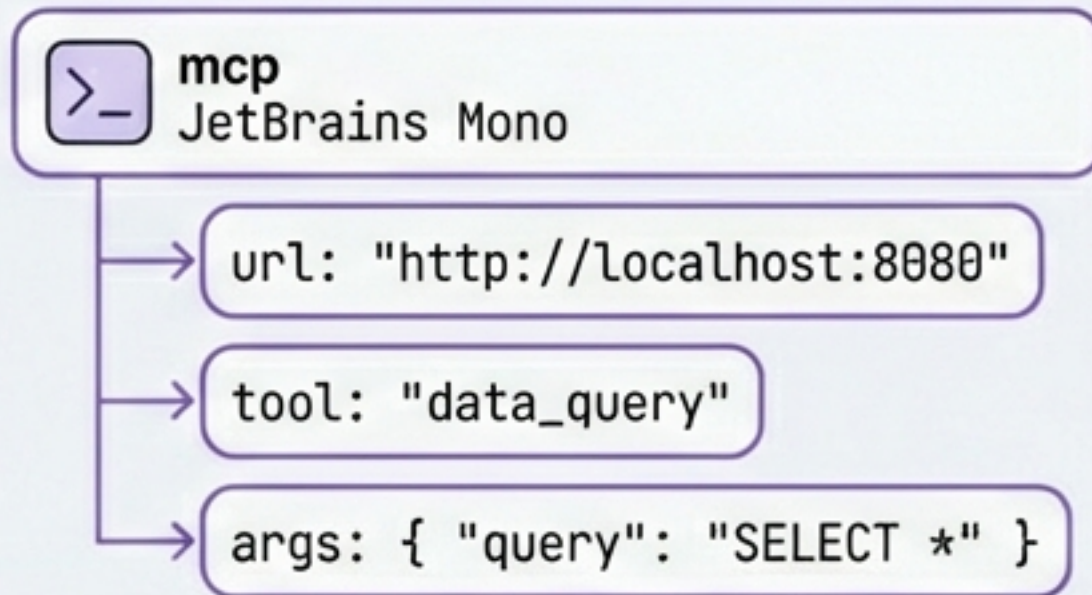


# Advanced Integration (MCP & Commands)

## Model Context Protocol (MCP)



Connect to external tools, databases, or local scripts.



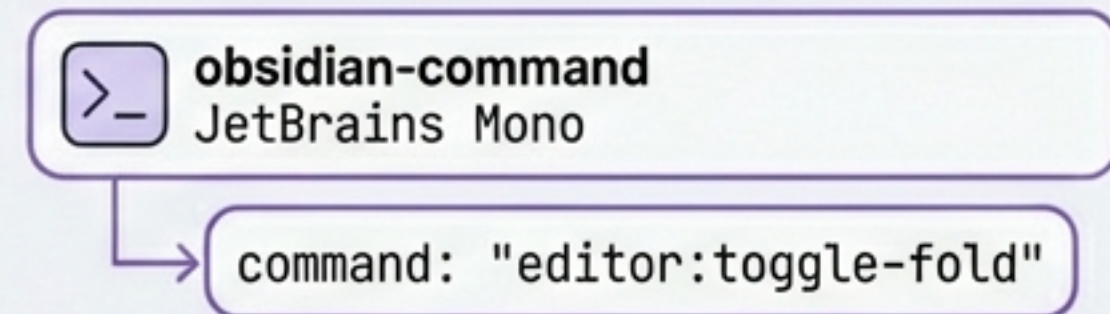
### Use Case

Standardizing AI tool usage across applications.

## Obsidian Commands



Trigger \*any\* Obsidian command ID within a workflow.



### Discovery

Find IDs in Obsidian Settings -> Hotkeys.



# Privacy & Data Architecture

## Local Device - Safe Zone



### Stored Locally

API Keys	✓
Chat History	✓
Workflow Definitions	✓

### Encryption

Supports End-to-End Encryption (E2EE) for logs.

### CLI Mode

Data piped to local child process (Google collection disabled for CLI).

## Cloud Services - Transfer Zone



### Sent to Google API

Chat Messages / Attachments	➡
RAG Index Files (Internal Mode Only)	➡

User has full control over what data leaves the device via RAG settings and Model selection.



# Quick Reference & Commands

## Key Commands (Palette)

- ▢ Gemini Helper: Open chat
- ▢ Gemini Helper: Show edit history
- ▢ Gemini Helper: Toggle chat / editor

## Workflow Hotkeys

Assign via Settings -> Hotkeys

- ▢ Search: 'Workflow: [Name]'

## Resources & Debugging

### **\*\*Docs\*\***

README.md &  
WORKFLOW\_NODES.md in repo

### **\*\*Debug Mode\*\***

Enable in Workflow settings to  
visualize variable states.

### **\*\*Repository\*\***

[github.com/takeshy/obsidian-gemini-helper](https://github.com/takeshy/obsidian-gemini-helper)