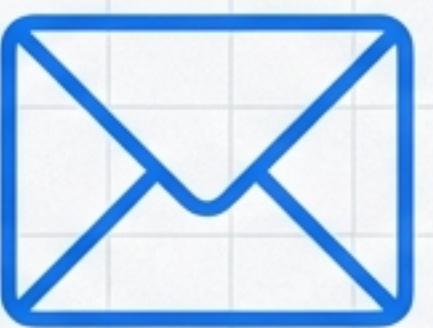


google-gmail-tool

A Professional Command-Line Interface for Google Services



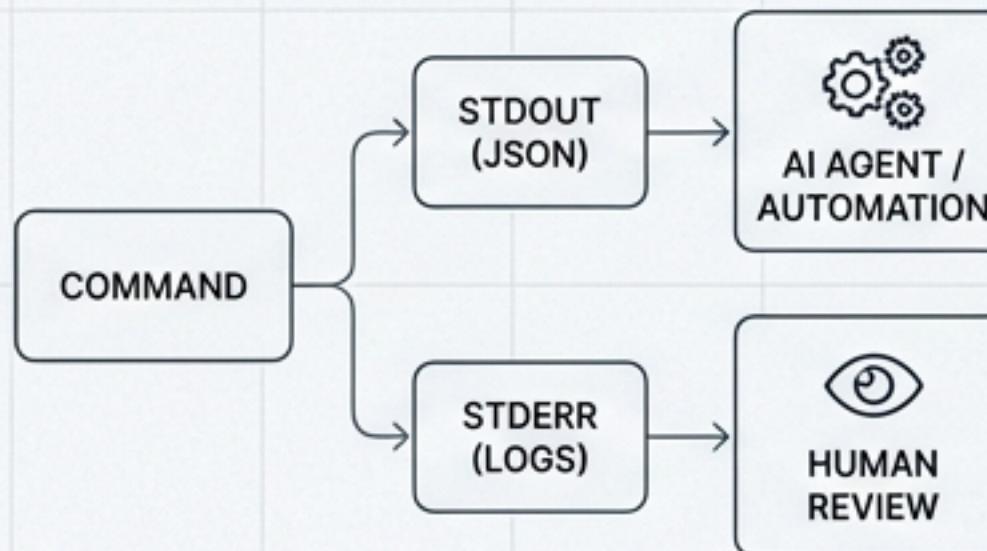
Note: This code was generated with assistance from AI coding tools and has been reviewed and tested by a human.

Engineered with Clear Design Principles



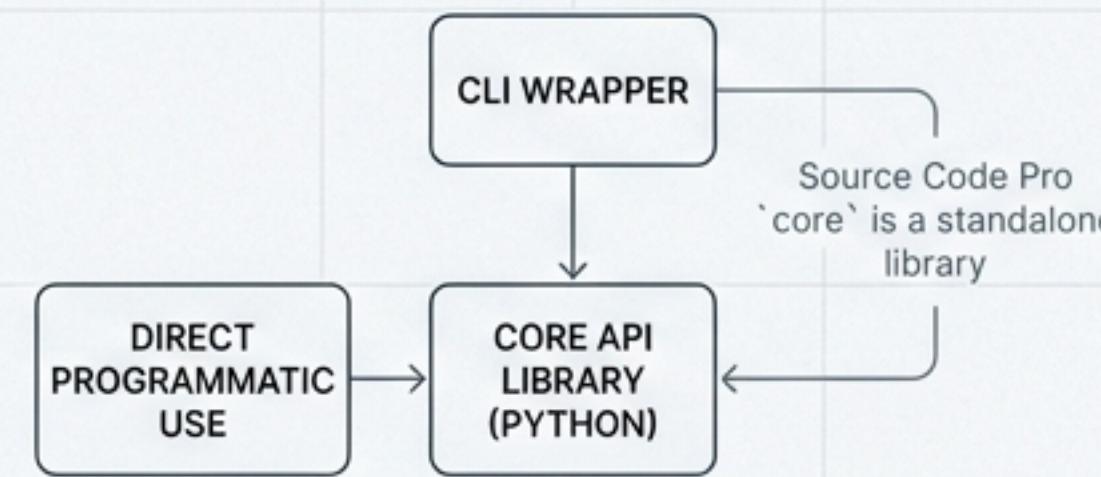
Agent-Friendly by Design

Structured for both human and AI interaction. Commands output machine-readable JSON to stdout and human-readable logs to stderr, enabling seamless integration into automated workflows and ReAct loops.



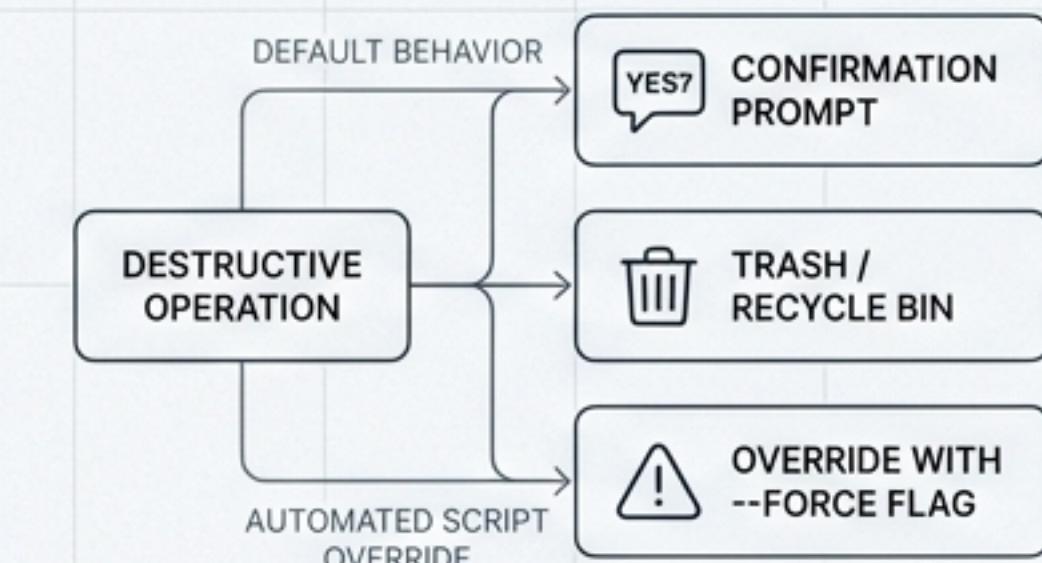
Decoupled & Reusable Core

A strict separation between the CLI command layer and the core API client library. The `core` is a standalone Python library, allowing for direct programmatic use without the CLI wrapper.

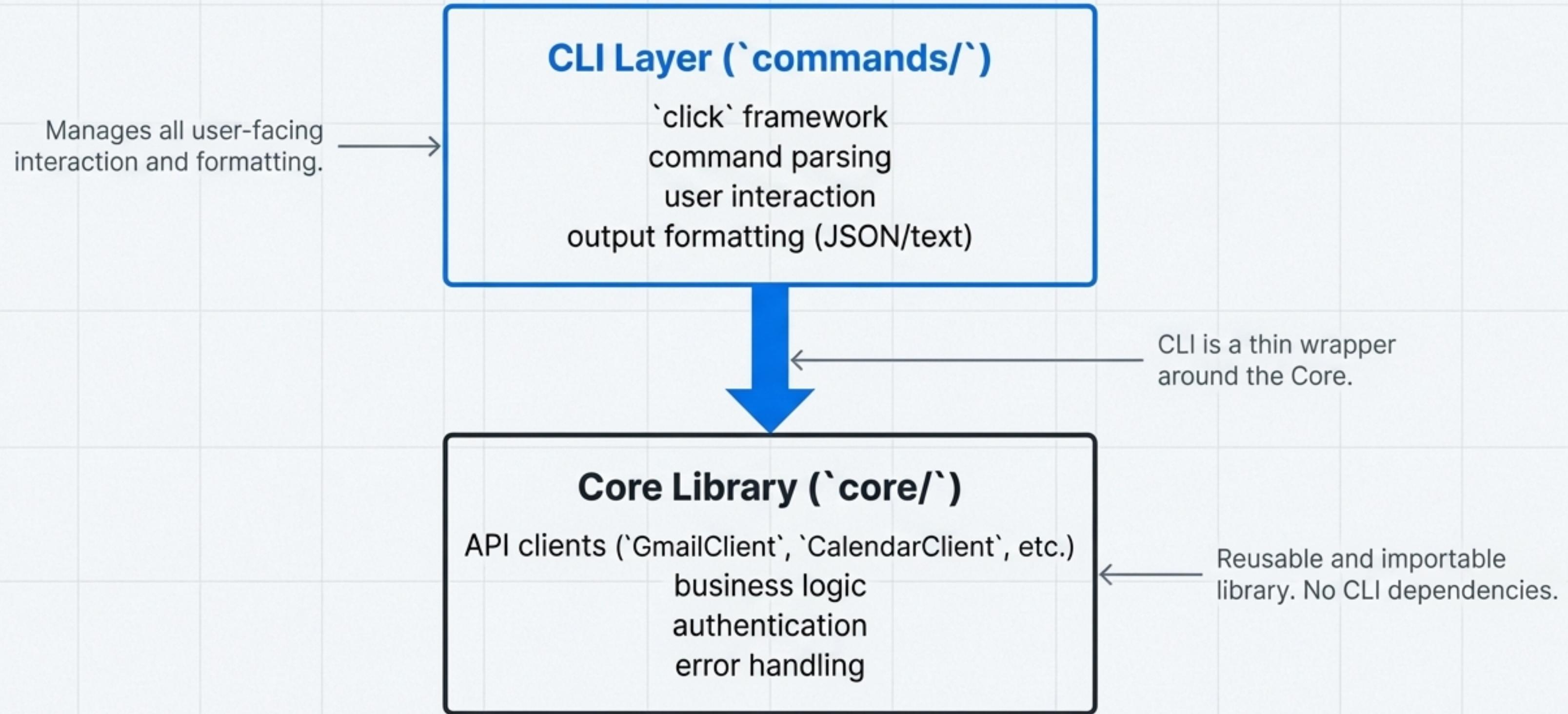


Safety First

Destructive operations are designed to be safe by default. Features include interactive confirmation prompts, trash-by-default for deletions, and `--force` flags for explicit override in automated scripts.



A Decoupled, Two-Layer Architecture



From Blueprint to Filesystem: The Project Structure

```
google_gmail_tool/
    ├── cli.py                      # Main CLI entry point
    └── logging_config.py          # Centralized logging and verbosity

    ├── core/                      # Core API Clients (CLI-Independent)
    │   ├── auth.py
    │   ├── gmail_client.py
    │   ├── calendar_client.py
    │   ├── task_client.py
    │   ├── drive_client.py
    │   └── obsidian_*.py           # Exporter logic

    └── commands/                  # Command Implementations
        ├── auth_commands.py
        ├── mail_commands.py
        ├── calendar_commands.py
        ├── task_commands.py
        └── drive_*.py
```

The file structure directly reflects the architectural separation, making the codebase predictable and easy to navigate.

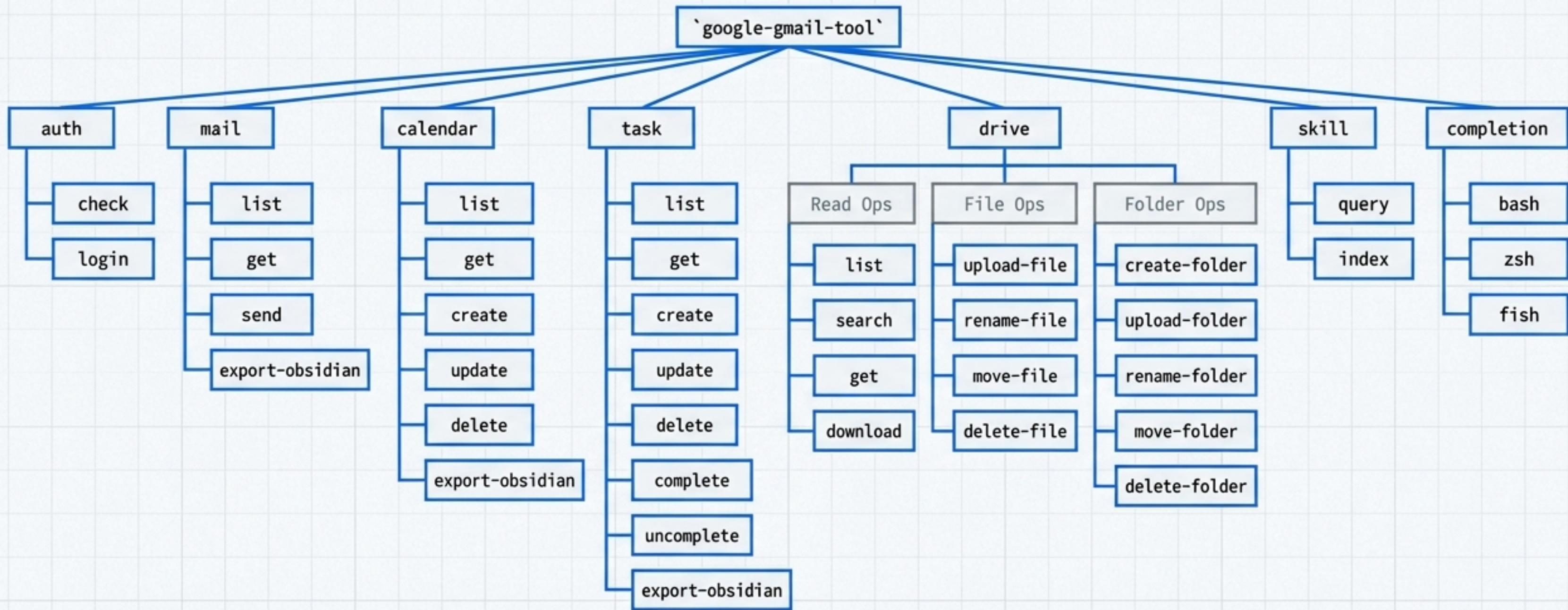
The Foundation: Authentication & API Scopes

The tool uses OAuth 2.0 for secure access. The initial `auth login` flow requests the following permissions to enable its full feature set:

Scope	Service	Access Level
<code>https://www.googleapis.com/auth/gmail.readonly</code>	Gmail	Read messages and threads
<code>https://www.googleapis.com/auth/gmail.send</code>	Gmail	Send emails
<code>https://www.googleapis.com/auth/calendar</code>	Google Calendar	Full access (read, create, update, delete)
<code>https://www.googleapis.com/auth/tasks</code>	Google Tasks	Full access (read, create, update, delete)
<code>https://www.googleapis.com/auth/drive</code>	Google Drive	Full access (read, write, delete)

Full access scopes for Calendar, Tasks, and Drive are necessary for the tool's create, update, and delete functionalities.

The Command-Line Interface at a Glance



A consistent and predictable structure ([GROUP] [COMMAND]) is used across all services.

Getting Started: The `auth` Command Group

Handles the initial OAuth2 flow and ongoing verification of API access.



`auth login`

Initiates a browser-based OAuth flow to obtain authorized credentials, including the essential `refresh_token`. Saves credentials to `~/.config/google-gmail-tool/credentials.json` by default.

```
# Easiest method, using the downloaded JSON
from Google Cloud
google-gmail-tool auth login --json-file
~/Downloads/client_secret.json
```

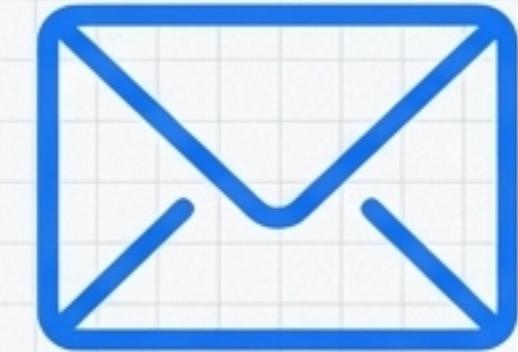


`auth check`

Verifies that credentials are valid and that the necessary API permissions have been granted for Gmail, Calendar, Tasks, and Drive. Provides clear success (✓) or failure (✗) feedback for each service.

```
# After login, verify the setup
export GOOGLE_GMAIL_TOOL_CREDENTIALS=~/.
conf
ig/google-gmail-tool/credentials.json
google-gmail-tool auth check
```

Core Capability: Managing Gmail with `mail`



Read, send, and export emails, leveraging the full power of Gmail's search syntax.

`list`

Lists messages or threads.
Supports full Gmail search operators via the `--query` flag.

```
google-gmail-tool mail  
list --query "is:unread  
has:attachment  
newer_than:7d"
```

`send`

Sends an email. Supports plain text, HTML, CC, BCC, and reading the body from a file.

```
google-gmail-tool mail  
send --to user@example.com  
--subject "Report"  
--body @report.txt
```

`export-obsidian`

Exports full email threads to structured markdown notes in an Obsidian vault, including frontmatter, attachments, and markdown-converted bodies.

```
google-gmail-tool mail  
export-obsidian --query  
"from:boss@work.com" -n 10
```



Core Capability: Interacting with Google Calendar via `calendar`

Full event management with powerful time-based filtering and smart Obsidian integration.

`list`

Lists events with flexible time-range flags (`--today`, `--this-week`, `--days 7`) and keyword search.

```
google-gmail-tool calendar  
list --this-week --query  
"standup"
```

`create`

Creates new events, with support for locations, attendees, and adding Google Meet links automatically.

```
google-gmail-tool calendar create  
--title "Remote Standup"  
--start "2025-11-20 09:00"  
--end "2025-11-20 09:30"  
--add-meet
```

`export-obsidian`

Exports events to daily notes as a checklist. Uses 'smart merge' logic to preserve checked items from previous exports.

```
google-gmail-tool calendar  
export-obsidian --today
```



Core Capability: Managing Google Tasks with `task`

Full task management including creation, status updates, and filtering.

list

Lists tasks with filters for completion status (`--completed`, `--incomplete`) and due dates (`--today`, `--overdue`).

```
google-gmail-tool  
task list --overdue  
--text
```

create

Creates a new task with a title, and optional notes and due date.

```
google-gmail-tool  
task create --title  
"Review PR #123"  
--due "2025-11-20"
```

complete

Marks one or more tasks as complete. Accepts multiple task IDs for batch operations.

```
google-gmail-tool  
task complete  
abc123xyz def456
```

export-obsidian

Exports tasks to daily notes, organized into sections like 'Overdue', 'Today', and 'This Week'.

```
google-gmail-tool  
task export-obsidian  
--this-week
```



Core Capability: Navigating Google Drive (Read Operations)

Comprehensive file discovery and retrieval.

list

Lists files using the full Drive API query syntax for advanced filtering by `mimeType`, `name`, `modifiedTime`, and more.

```
google-gmail-tool drive  
list --query "name  
contains 'invoice' and  
mimeType='application/p  
df'"
```

search

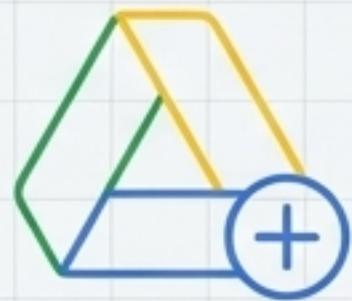
A simplified search interface with common filters for name, MIME type, and folder.

```
google-gmail-tool drive  
search --name "budget"  
--mime-type  
"application/vnd.google-  
apps.spreadsheet"
```

download

Downloads a file by its ID to a specified local path. Handles binary files directly.

```
google-gmail-tool drive  
download "1abc123xyz"  
"/tmp/report.pdf"
```



Core Capability: Managing Google Drive (Write Operations)

Full file and folder management with an emphasis on safe, predictable operations.

`upload-file` / `upload-folder`

Uploads single files or entire folders recursively. Checks for duplicates and prompts for confirmation unless --force is used.

```
google-gmail-tool drive  
upload-folder  
"/path/to/project"  
"/path/to/project"  
"1parentFolderID"
```

`move-file` / `move-folder`

Moves a file or folder to a new parent location.

```
google-gmail-tool drive  
move-file  
move-file "1abc123xyz"  
"1destination456"
```

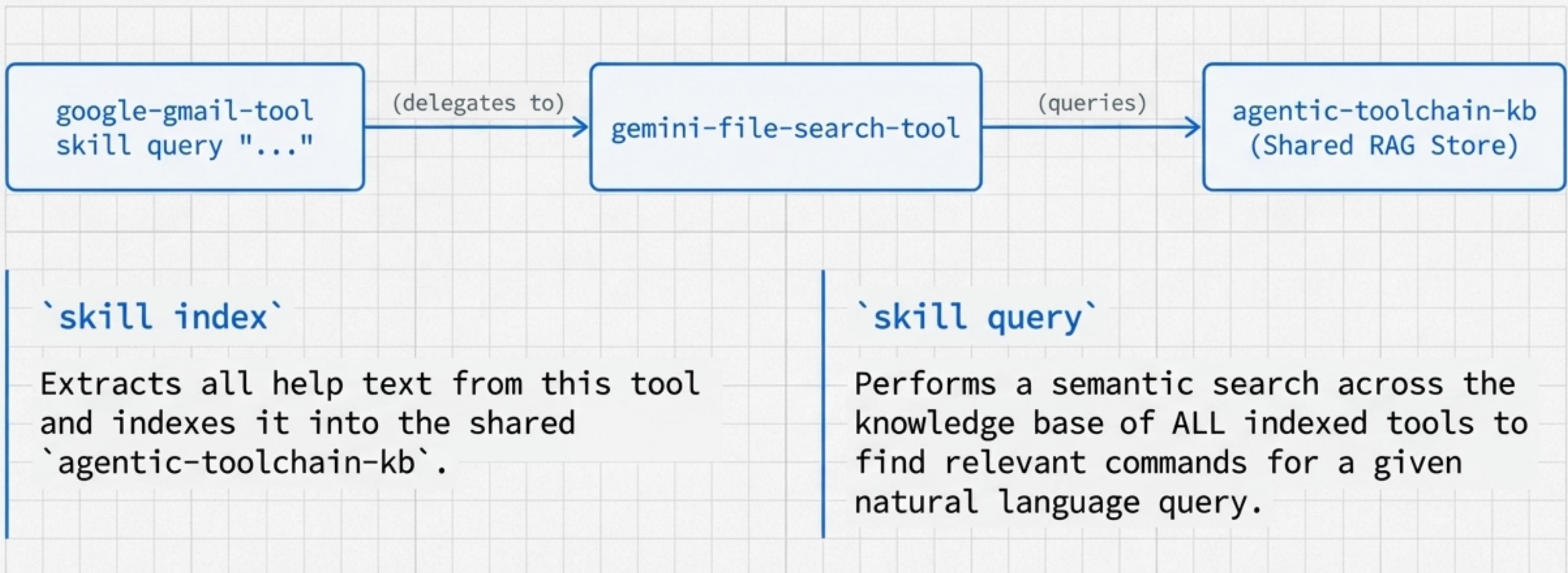
`delete-file` / `delete-folder`

Moves items to the trash by default. The --permanent flag is required for irreversible deletion, and a confirmation prompt is shown unless --force is used.

```
google-gmail-tool drive  
delete-folder  
"1oldFolderID" --force
```

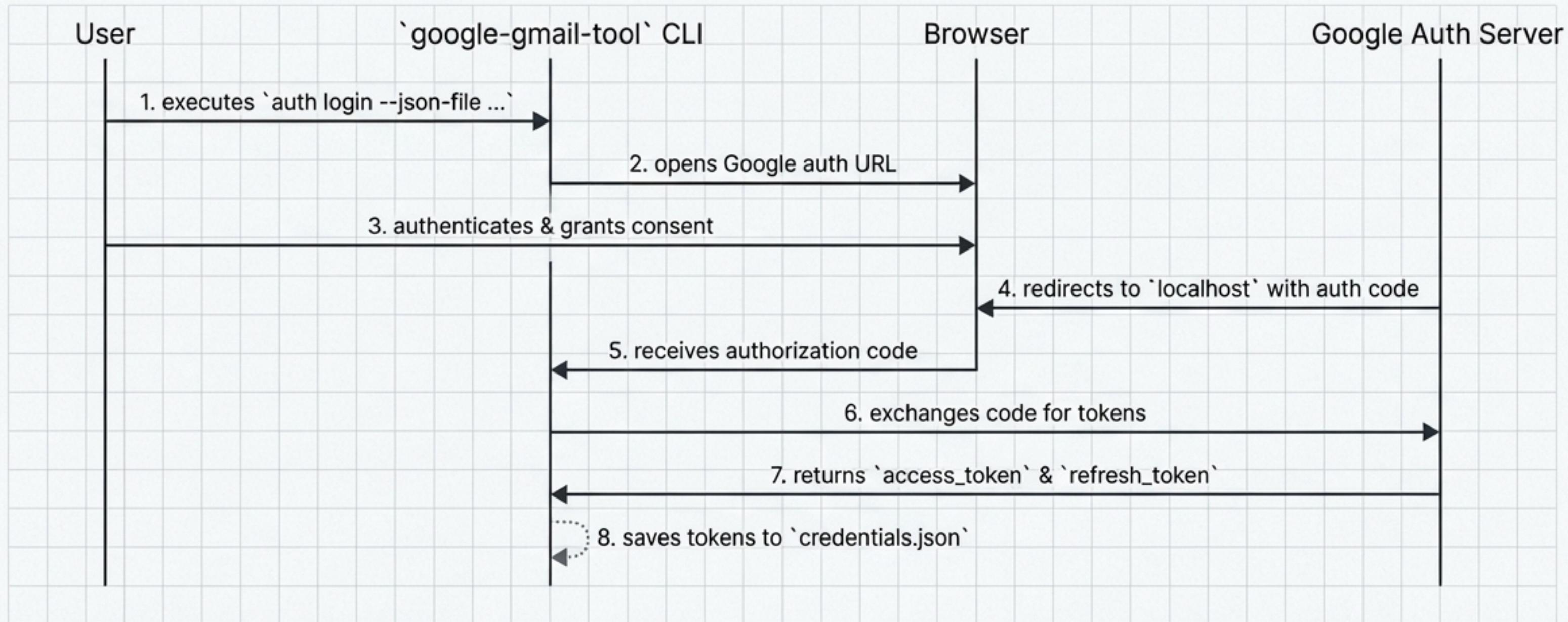
Beyond Human Interaction: The `skill` Command Group

Implements the 'CLI-as-Skill' pattern, enabling AI agents to discover this tool's capabilities through semantic search.



This positions `'google-gmail-tool'` as a composable skill within a larger, intelligent toolchain, discoverable by AI agents.

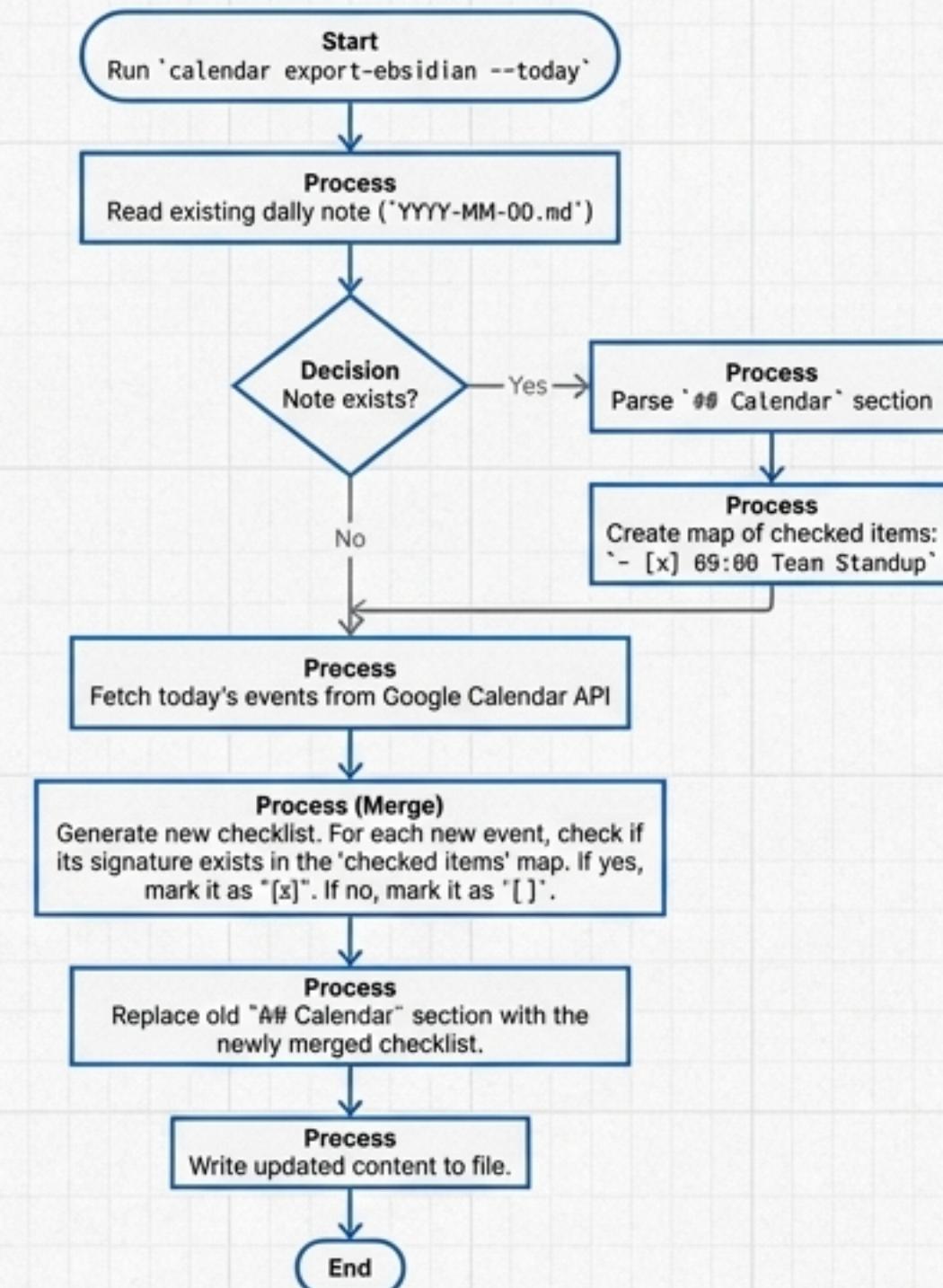
In Action: Visualizing the `auth login` Flow



This flow securely obtains a long-lived `refresh_token`, allowing the tool to operate non-interactively in the future.

In Action: The Obsidian Smart Merge Logic

How `calendar export-obsidian` updates your daily note without overwriting your completed tasks.



The Result: Structured Knowledge in Obsidian

mail export-obsidian

```
---  
subject: "Project Update Q4"  
from: "boss@work.com"  
tags: [email, gmail/inbox]  
---
```

Project Update Q4

Here is the latest update...

- [[report.pdf]]

calendar export-obsidian

```
## Calendar  
  
- [x] 09:00-10:00 Team Standup  
@ Zoom  
- [ ] 14:00-15:00 Client  
Meeting @ Conference Room A
```

task export-obsidian

```
## Tasks  
  
### Overdue  
- [ ] Task 1 (due: 2025-11-15)  
  
### Today  
- [ ] Task 2 (due: 2025-11-20)
```

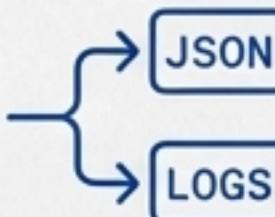
A Deeper Dive: Principles of Agent-Friendly Design

The tool's design is guided by patterns that enable reliable use by AI agents and automation scripts.

Clear I/O Channels

Structured `JSON` data is written to `stdout` for easy parsing.

Human-readable logs, progress, and errors are written to `stderr` to avoid polluting the data channel.



Self-Documenting Help

Every command includes comprehensive `--help` text with progressing examples, from simple to complex, allowing an agent to learn its capabilities and syntax.



Predictable Exit Codes

Consistent exit codes signal outcomes: `0` for success, `1` for general errors, and `2` for 'not found' or user cancellation. This allows scripts to reliably handle different results.



Idempotent & Non-Interactive Flags

Flags like `--force` and `--auto-approve` bypass interactive prompts, ensuring that scripts can run non-interactively without hanging.



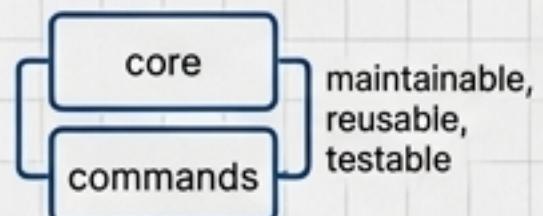
A Critical Perspective: Strengths

Based on a professional analysis of the tool's design and features.



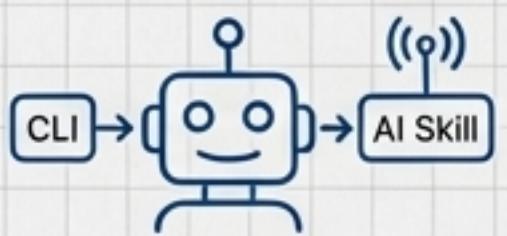
✓ Comprehensive Service Coverage

Integrates the four most common Google productivity services (Mail, Calendar, Tasks, Drive) under a single, consistent interface.



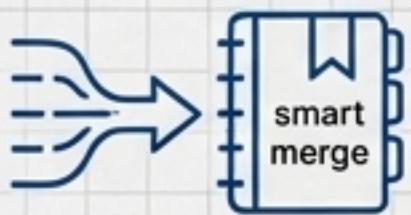
✓ Robust Architectural Design

The strict separation of the `core` library from the `commands` layer ensures code is maintainable, reusable, and testable.



✓ Advanced Agentic Features

The “CLI-as-Skill” concept with the `skill` command group is a forward-thinking feature that positions the tool for future AI-driven ecosystems.



✓ High-Value Integrations

The Obsidian export functionality with “smart merge” logic provides unique value for users in the knowledge management space.



✓ Focus on Safety and Usability

Features like trash-by-default, duplicate detection, and clear confirmation prompts make the tool safe and reliable for both interactive and automated use.

A Critical Perspective: Limitations & Opportunities

Continuing the professional analysis.

Limitation

Google Workspace file formats (Docs, Sheets) cannot be downloaded directly, only binary files.



Limitation

The powerful `skill query` functionality depends on an external tool (`gemini-file-search-tool`).



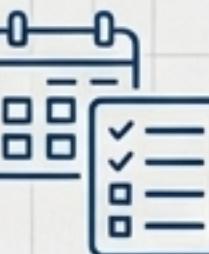
Limitation

Task searching (`task list --query`) is performed client-side after fetching all tasks.



Limitation

The tool currently operates on the default task list and primary calendar only.



Opportunity

Implement an `export` command for Drive files to convert and download them in formats like PDF or XLSX.



Opportunity

Future versions could explore embedding a lightweight RAG (Retrieval-Augmented Generation) model directly into the tool for a more self-contained experience.



Opportunity

Enhance the `TaskClient` to support more advanced server-side filtering if the Google Tasks API allows, improving performance for users with many tasks.

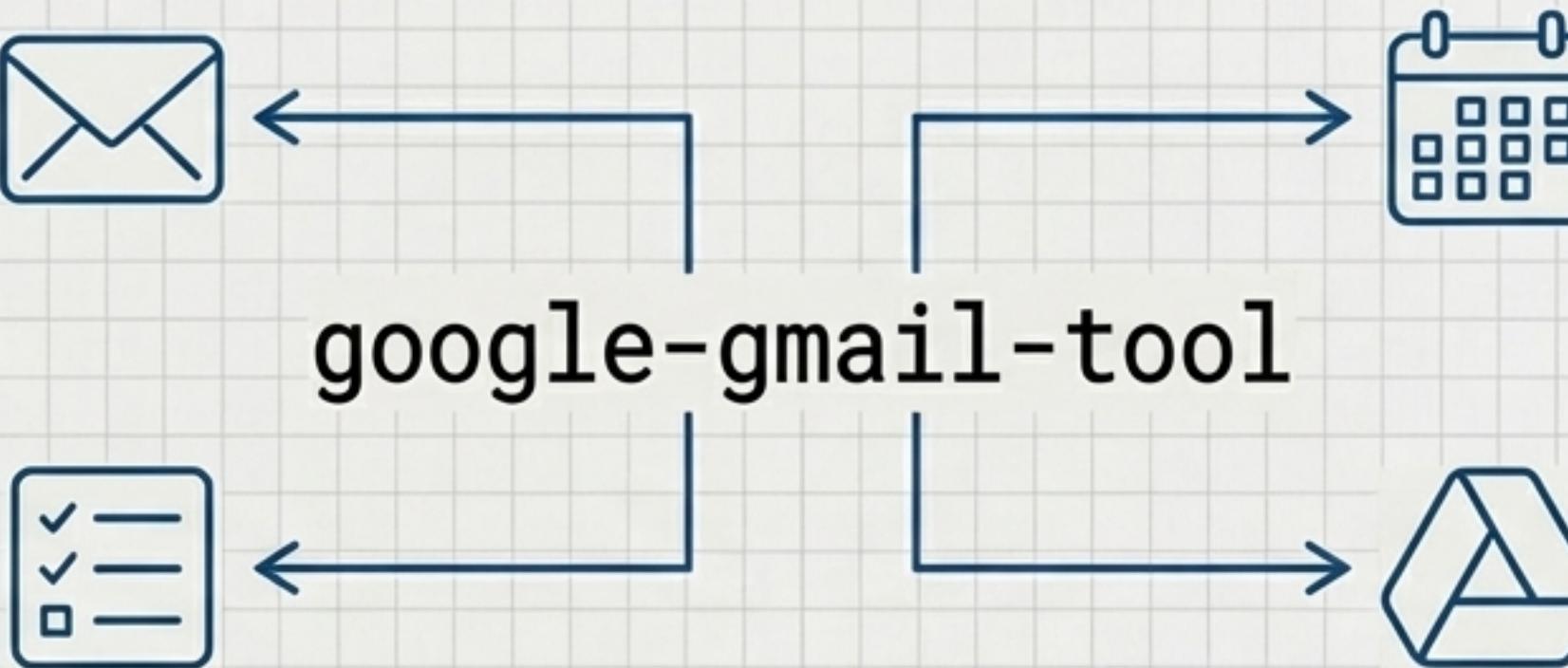


Opportunity

Add options to specify a task list ID or calendar ID, allowing users to manage multiple lists and calendars.



A Professional, Agent-Ready CLI for Your Google Workspace



`google-gmail-tool` is more than a utility; it's a well-engineered, reusable, and intelligent component for modern development and automation workflows. It provides a robust and safe interface to essential Google services, designed for both today's power user and tomorrow's AI agent.

Get Started

GitHub Repository

github.com/dnvriend/google-gmail-tool

Installation Command

```
# From the cloned repository  
uv tool install .
```