



// SECTION 01 • OPEN SOURCE

MUESLI

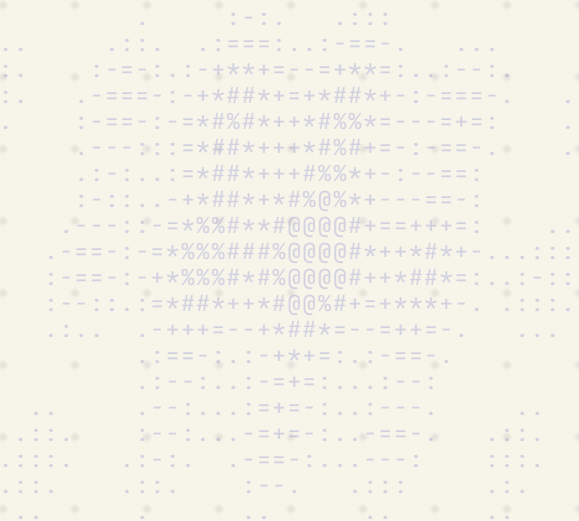
FIELD GUIDE



An open-source Granola-style meeting notepad for macOS. No bot joins your calls, transcription runs on your own machine, and your rough notes come out polished. This is the clone-to-first-note guide.

```
> boot opusjake_os
> resource: muesli
> version: v1.0 • 2026
> status: ready_
```

I use AI meeting notes every day, and I wanted the whole pipeline on my own machine: the audio, the transcript, the notes. So I built Muesli, an open-source Granola-style notepad for macOS, and put the code on GitHub. It records both sides of any call without a bot joining, transcribes locally with Whisper, and rewrites your rough notes into real ones with Claude or a local model. This guide takes you from clone to your first enhanced note in about ten minutes.



WHAT IT IS

Muesli is a native macOS app. One Swift package, zero dependencies beyond the whisper.cpp binary, MIT licensed. It does four things.

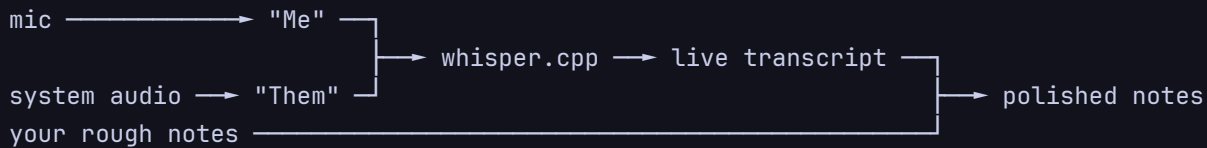
- **Records without a bot.** Your mic is one stream, everything your Mac plays is the other. Zoom, Meet, Teams, a phone call routed through the Mac, all of it. Nobody sees "Muesli has joined the meeting" because nothing joins.
- **Transcribes locally.** whisper.cpp runs on your machine in roughly 20 second chunks while you talk. Your mic is labeled Me, the other side is labeled Them.
- **Enhances your notes.** You type fragments during the call. Afterwards, one button merges them with the transcript into polished notes: decisions, action items, open questions.
- **Stores plain files.** Every meeting is a folder with meeting.json, notes.md, transcript.md, enhanced.md, and the WAV audio. grep works. Nothing is locked in.

	GRANOLA	MUESLI
Price	Subscription	Free, MIT license
Source code	Closed	github.com/Jakeschincariol/muesli
Transcription	Cloud	whisper.cpp on your Mac
Audio and transcripts live	Their servers	A folder in your Library
Notes model	Theirs	Your Claude key, or Ollama fully local
Bot in the call	No	No

Granola is a good product. This is the same breakfast with every ingredient visible.

HOW IT WORKS

The trick that makes bot-free capture possible is ScreenCaptureKit, the macOS framework that powers screen recording. It can tap system audio, meaning everything the Mac plays. That tap is the other side of your call.



- **Chunking.** Audio is cut every 20 seconds or so, at moments of silence so words do not get split. Each chunk of each stream runs through Whisper and lands in the transcript, interleaved by timestamp. On Apple Silicon the recommended model transcribes 10 to 20 times faster than realtime.
- **Enhancement.** Your rough notes set the emphasis, the transcript supplies the facts. The prompt tells the model to keep your structure, fix names and numbers from the transcript, and pull out what you missed. Six templates change the output shape: Auto, 1:1, Sales, Standup, Interview, Brainstorm.
- **Privacy line.** Recording and transcription never touch the network. The only optional network call is enhancement, and pointing Muesli at Ollama removes even that.

INSTALL IT

You need an Apple Silicon Mac on macOS 14 or newer, plus Homebrew.

```
brew install whisper-cpp
git clone https://github.com/Jakeschincariol/muesli.git && cd muesli
make run
```

First launch shows a five item checklist. Two minutes, once.

- **Microphone.** One click.
- **System audio.** macOS files this under Privacy and Security, Screen and System Audio Recording. Enable Muesli there, then relaunch. Audio-only capture still lives behind the screen recording permission; that is an Apple decision, not mine.
- **Whisper engine.** Found automatically after the brew install.
- **Speech model.** One click downloads base.en, 142 MB, one time.
- **AI provider, optional.** Paste an Anthropic API key (stored in the macOS Keychain) or point it at Ollama. Only enhancement and Q&A need this. Recording and transcription work without it.

Then press Start meeting, talk, type fragments, press Stop, press Enhance.

USE IT WELL

- **Type badly on purpose.** Rough notes are a steering wheel, not a record. "pricing?? sarah owns redesign" is plenty. The transcript carries the facts.
- **Wear headphones on speaker calls.** Muesli separates Me from Them by audio source. Headphones keep your mic from hearing the other side twice.
- **Pick a template before you enhance.** A sales call and a standup produce different notes from the same transcript.
- **Ask the meeting questions.** Each meeting has a chat tab grounded in its transcript. "What did we decide about pricing" and "draft the follow-up email" both work.
- **Re-transcribe when you upgrade models.** The full session audio stays on disk. Swap in a bigger Whisper model later and rebuild the transcript from the original WAVs in one click.

MAKE IT YOURS

The codebase is small on purpose. Four folders matter.

FOLDER	WHAT LIVES THERE
Audio/	Mic capture, system audio tap, chunking, WAV writing
Transcription/	whisper.cpp wrapper, job queue, model downloads
Enhance/	Claude and Ollama clients, note templates
UI/	SwiftUI: sidebar, transcript, notes, settings

Things I want to exist and have not built yet: calendar awareness so meetings auto-start and auto-title, an importer for your Granola history, search and chat across every meeting at once. If one of those is your itch, the repo takes pull requests.

Clone it tonight, record one real call tomorrow, and read the enhanced notes before you read the transcript. That order sells the whole idea. Code at github.com/Jakeschincariol/muesli, more builds at opusjake.ai.