

The Door Table and the Moving Twig: Replayable Installs and the Stick Bug Secret

Setting the Stage: Context for the Curious Book Reader

Context for the Curious Book Reader

In this installment of the living book, we hit a practical and philosophical crossroads in software independence. What begins as a gritty, power-outage-punctuated sprint to polish an automated onboarding flow turns into an architectural breakthrough: decoupling an open-source workshop's installer so a single script can cleanly serve multiple web doors (pipulate.com, npvg.org, and qamy.ai) without creating messy codebase forks.

Along the way, the entry articulates the *myelination ladder*—the neurological process by which external tools become internalized reflexes, moving from hesitant literacy to automaticity, prosody, and permanent fluency. While the mainstream tech ecosystem profits from perpetual churn and fragile web interfaces that reset users to square one, this methodology anchors itself in reproducible, local-first environments and verified receipts. Underneath the unpretentious, quiet CLI menu lies "Operation Stick Bug" (osb): a hidden layer of live telemetry disguised as dull checklist furniture, paired with an Achilles and the Tortoise dialogue that proves why running real code beats arguing with an AI's hallucinations every single time.

TL;DR: Pipulate is a workshop that installs itself on a user's own computer with one command. One install script is served from three web addresses, and each address now gives the install its own folder name and its own opening sentences. The web server rewrites one placeholder word in the script as it serves it. The script uses that word to pick a row in a small table, creates the folder that row names (~/.qamyai for qamy.ai), and saves the row's sentences there. On first start the program prints those sentences. It reads them aloud with an offline text-to-speech voice only if the user agrees. A hidden command, osb, prints live system readings and short stories labeled as stories. Each change was checked by running the same commands before and after it, on Linux and on a Mac. The audio itself, the other two addresses and the Lix variant of Nix are not yet checked.

Technical Journal Entry Begins

 Verified Pipulate Commits:

- [43006cba](#) ([raw](#))
- [3f09904d](#) ([raw](#))
- [46118ec0](#) ([raw](#))

- [34bbaa4b](#) ([raw](#))
- [868425bd](#) ([raw](#))
- [869637dc](#) ([raw](#))
- [daaac215](#) ([raw](#))
- [8149a74c](#) ([raw](#))
- [5dce5740](#) ([raw](#))

MikeLev.in: There is something new under the Sun and you can't prompt it from out of AI. No models have been trained on this. Like they say, if it hasn't been trained on from GitHub or Stack Overflow then the AI can't spontaneously create it. It's not creative that way. It's a token predictor and while the combinatorial inventiveness is potentially unlimited, finding an actual new way to skin that cat is not.

Here we have written a Context Compiler and system to Q/A your AI that cannot exist except by borrowing the work directly from another human because the AIs aren't this smart yet.

They can't do it.

If they could, everyone would be doing it and working this way instead of all paths leading to paying someone for some service and locking you into some vendor, locating your work where you don't own it and if you stop paying it's gone and the easiest thing to do is to scrap-and-rebuild which makes them the most money anyway, keeps you on the hamster-wheel and prevents you from...

...the word is **myelinating** the skills.

I would explain this word to you but I haven't gotten over the hump with good enough communication skills to explain it yet.

I'm just not fluent enough in the internalized tool we call "language".

I'm not expressive enough yet.

Sure, sure, I can read and write and blog an blah blah blah well enough.

Of course I can because I'm literate, but literacy is just the start of such things as being able to ride a bicycle where you internalize external tools such that they're like an extension of your body and you don't have to deliberately think about them.

There's these experiments where they reverse the bicycle controls. People do terribly at first but then they get a hang of it and become literate at this wacky turn-right go-left, turn-left go-right dynamics.

Humans can adapt to that kind of shit but you know what happens when they try to go back to a normal bicycle? I bet you can guess.

They have to become basic literate again.

Forget it being automatic. When it becomes automatic and you don't have to think about it anymore that's actually a bit more than literate. If you have to stop and think and use your executive function really figuring something out, you can get by. You're literate but without that automaticness that the funny word for it is automaticity.

That's what we call muscle memory.

Black gold.

Texas tea.

Myelination.

The next thing you know, poetic you will be just flinging it together. That word is prosody.

It's beyond just mere literate and automaticity.

Keep the pedal to the metal and develop fluency!

The Four Stages of Myelination: From Literacy to

Fluency

The meta-ness here is a direct homage to a book called Strunk and White, *The Elements of Style*, a famous short guide to English writing that does this silly kind of stuff too...

...but way shorter and much more concise and unusing of made-up words and gratuitous needless verbosity and over-erudition of exactly the type I keep slamming Claude for using at me but I know it's just ***magic mirroring*** my own style but it's fine when I do it but I don't want my AI doing it where I actually have to understand what it's saying and follow steps.

Here's what's currently at the top of my super-duper-incredibly important apply.py file in this system:

How the User wants the Model to write and work for him; read before drafting anything or handing him commands

- Dislikes ornate, over-erudite model prose; calls it "H.M. Wogglebug language" or "PhD language" and asks Claude to speak plainly
- Wants answers short and dry, minimal words, the least reading effort needed to digest them
- Says he misses any action item for him that is phrased politely or buried in prose; wants anything he must do stated plainly and made impossible to miss, separate from the explanation
- Never create Artifacts (out-of-band pages); findings go in the main response where he can copy them
- Use rg, never grep, in shell commands for him, because .git/ folders are everywhere
- Treats unreproduced claims as confident hallucination until reproducible; wants provenance for technical claims: endpoint, parameters, auth method, account, timestamp, and local file plus search string
- In Prompt Fu sessions, end every response with the 5-Car Train: PROBES (read-only, runnable on his Mac), NEXT CONTEXT (paste-ready adhoc.txt lines, never prose), PATCHES (real SEARCH/DIVIDER/REPLACE blocks for apply.py), PROMPT caboose, EXTERNAL DELIVERABLES
- Never wants to go on scavenger hunts; needed material should live in his
- corporate and personal repos

Mnemonics are important.

I just made the 3 key commands of the new main thrust of the system (since operation stick bug):

1. prompt
2. context
3. compile

...able to be typed with:

1. p
2. x
3. c

...which sort of sounds like pix-see or pixie.

And installing Nix with the DetSys installer (I just learned about lix and more on that in the future) is a lot like sprinkling magic pixie-dust on your computer and having this Flight Deck Recorder blackbox system *outside* the browser which makes all the difference and also gives us this wonderful mnemonic.

Using any one of these systems:

1. qamy.ai
2. npvg.org
3. pipulate.com

...is like sprinkling magic pixie-dust on your computer unlocking tools which over time you will:

1. become literate with
2. develop muscle memory for (automaticity)
3. slam together like it's no big (prosody)
4. carry forward for life (fluency)

But this won't happen with me making a single site with a single bit of messaging. Because each unit of messaging and each try I make at getting this software Von Neumann probe bird off the ground (the wind-tunnel tests at Dayton Ohio) has to be tight, tight, tight, I have to make many tries over and over and over, iterating through messages and domain names and gists until the right ones stick with the right audience.

And no matter which site, it's the system where you sprinkle the magic pixie dust on your machine and get the AI Prompt conteXt Compiler and the commands: p, x, c which sounds like pix-see which sounds like pixie. And then with that core I iterate fast with different versions using different domain names and allow each person using this system to fork it, white-label it and do the same providing safe harbor for SEOs looking for the next big thing among all the noise that's not just vibe-coding your future tech liability.

The Multi-Door Architecture: Decoupling Domains from Directory Names

In other words, Pipulate should be able to be known by N different names and it should not be so exactly tied to the apex domain minus TLD the way it is now with the pipulate.com install going into pipulate/, the npvg.org install going in npvg/ and especially the qamy.ai install going in qamy/. It's much more powerful to say:

"Now that you've installed QA my AI go to the directory at the root of your machine's user home folder called qamyai," which sounds when spoken like "QA my AI" which is a lot more powerful than what comes off sounding like "commie". And while we're at it we want something different read immediately after Piper speaks as well:

This workshop can read its steps aloud with Piper, a small voice that runs on this computer.

Downloading the voice, about 60 MB, once...

Voice ready.

This is Piper, reading a script. If you can hear this, press Enter.

Type n to keep it quiet.

Enter or n:

Voice on. Type talk any time to turn it off.

But it should when installed from qamy.ai:

This LLM Quality Assurance workshop talks you through steps.

Voice ready.

This is only text-to-speech. No local AI is installed yet.

If you can hear this, press Enter. Type n to keep it quiet.

Enter or n:

Voice on. Type talk any time to turn it off.

Welcome. Ready to Q/A your AI with the Flight Data Recorder (FDR)
Different from Cockpit Voice Recorder (CVR) or "what Claude said".

But if Ollama is detected as installed it should say:

This LLM Quality Assurance workshop talks you through steps.

Voice ready.

This is only text-to-speech. Local AI detected but not used.

If you can hear this, press Enter. Type n to keep it quiet.

Enter or n:

Voice on. Type talk any time to turn it off.

Welcome. Ready to Q/A your AI with the Flight Data Recorder (FDR)

Different from Cockpit Voice Recorder (CVR) or "what Claude said".

The install in this case should be to qamyai/.

And this is a messaging opportunity. I want to propose and pitch this system as countless different things only just starting with Pipulate, NPvg and QA my AI and each one should have its own critical few first words, so when installed from NPvg.org the messaging should say:

This Local-AI Experimental workshop talks you through steps.

Voice ready.

This is only text-to-speech. No local AI is installed yet.

If you can hear this, press Enter. Type n to keep it quiet.

Enter or n:

Voice on. Type talk any time to turn it off.

Welcome. Ready to raise Chip O'Theseus, your Local AI Hermit Crab

That moves with you from hardware to hardware without the Cloud.

The install here should be to pipulate/ or whatever they white-labelled which is the case on all these install sites, really though we might advertise that fact more or less from the install page of each version. Pipulate will get it advertised more because it's SEOs looking for safe harbor.

If it's installed from npvg.org I want:

This Future-Proofing workshop talks you through steps.

Voice ready.

This is only text-to-speech. No local AI is installed yet.

If you can hear this, press Enter. Type n to keep it quiet.

Enter or n:

Voice on. Type talk any time to turn it off.

Welcome. Ready to teach you battle-hardened timeless tech

Helping you stay relevant and a valuable voice at the table.

I imagine some sort of dictionary in the installer maybe? I'll be fine-tuning it over the years adding more and more installer surface-area, target audiences and the like. I don't want to be too limited but also I don't want a big messy system. A nice tight fill-in-the-blanks template system is fine.

I think I want one more custom tag-line per site though it may remain unused until we figure out where it belongs, but make this as part of the dictionary or whatever we end up using.

- **qamy.ai**: Treat claims as confident hallucination until reproduced.
- **pipulate.com**: Assume models guess at homepages and crawl it from there.
- **npvg.org**: Reproduce it with recipes or the system rots; no in-between.

Does that make sense?

Repeat everything I said back in your own words but in another way for a different kind of learner. First be super-brief and concise for people who like (and indeed, **need**) it stated more plainly than my long rambling norm.

And this is how the living book is born.

It's a lot like the powder of life in The Wizard of Oz books.

You think you know the Wizard of Oz because you saw the movie like a zillion times, but if you haven't read those first... what is it? 13 Baum-books, you don't know Oz.

Do you know about walking around the invisible wall protecting the land of the Skeezeres and its relationship to APIs and the behavior you have to have towards them?

Are you letting what you do with AI be guided by the Central Finite Curve of the normie hamster-wheel of tech where vendors hold the corporate world on a dependency leash?

Yeah, fish can't see water and the ancient Greeks didn't have a word for blue even though they could see it just fine, and this is for economic reasons that there is no word until somebody can make money off it. It's a Eric Raymond *The Cathedral and the Bazaar* scratching an itch thing, and if you don't know what I mean by that then it's an example of what I'm talking about.

And then be pedantic with your vetting and expanding and adding flavor and polish. I can host unlimited sites from my wee little retired Windows 10 laptop turned into a little recipe slave-drone of Pipulate Prime. I think you can see that from the 40K-foot view. Have at it and go for broke using 'dem tokens. MAXIMUM EFFORT!

Note: This ran for over an hour and as expected the thinking process where all the portable context of tool-calls in the background would require expanding too much for it to be practical.

Opus 5.5 Max: I'll rehearse the installer change against a copy of install.sh in my own sandbox and check three facts before writing the train.

I'll check three more facts I plan to cite, then rehearse the installer change in my sandbox.

One page I couldn't read: pubmed.ncbi.nlm.nih.gov/25324370 (a summary of the 2014 myelin study). The fetch proxy refused it as rate-limited. I'm going on without it and will cite that study from the search results instead.

Context verified for the installer half. Everything it touches is in this payload: install.sh, flake.nix's first-entry sweep, .gitignore, apply.py, the router and the glossary. Two files for the other half are missing, so I didn't patch them this turn. Both are in NEXT CONTEXT: - imports/voice_synthesis.py: it owns the voice card. - remotes/honeybot/www/qamy.ai/index.html: it may still tell people ~/qamy.

You

- **Don't publish yet.** No release and no ./nixops.sh --installer until the voice card and the qamy.ai page card land next turn. A git push doesn't change any door, so nobody installing will see a difference.
- **Edit the five rulings in the caboose before you send it.** My picks are filled in.

Does it make sense?

Yes. One thing to confirm: I read your second "NPvg.org" block as **pipulate.com**. It has the Local-AI Experimental line, Chip O'Theseus, installs to pipulate/ and talks about SEO safe harbor. I built it that way. I also read "npvg.org" as npvg.org.

The short version

1. AI remixes what it has seen. A working method nobody has published has to come from a person, and this is one.
2. Vendors make more money when you rebuild. Rebuilding keeps you a beginner.
3. Skill has four steps: - you can do it (literate) - you can do it without thinking (automatic) - you can do it with style (prosody) - you keep it for life (fluent)

The brain insulates practiced paths with myelin. Churn throws that away. 4. Three commands: p prompt, x context, c compile. Said together it sounds like "pixie", and installing Nix is the pixie dust. 5. One system has many names and doors. Each door gets its own first words, tested until they stick, the way the Wrights tested wings. 6. The folder isn't the domain name. qamy.ai installs to ~/qamyai, which sounds like "QA my AI". 7. Each door is one fill-in-the-blanks row: folder, first line, two welcome lines and a spare tagline.

The pedantic pass

Corrections - The wall guards the Flatheads, not the Skeezers. It's in *Glinda of Oz*, chapter 5. The Flatheads built it in front of the stairway into their mountain and made it invisible, "believing that everyone who saw the entrance to the mountain would walk straight toward it." - Ozma doesn't push. She "began feeling her way along the wall." After nearly a quarter mile it "curved in toward the side of the mountain and suddenly ended, leaving just space enough between the wall and the mountain for an ordinary person to pass through." - The gap is there because the Flatheads use those stairs themselves. - **The API lesson:** don't push on the front door. Follow the wall to where the builders' own traffic gets through. The vendor's own app has to reach its own data, so its network traffic (your "wire truth") shows you the gap. - Your glossary says "push through and it snaps." Car 2 fixes that. - **Baum wrote 14 Oz books, not 13.** *Glinda of Oz* (1920) is #14, and it's the one with the wall. - **The backwards bike didn't send Destin back to zero.** It took him eight months to learn and his son two weeks. Going back to a normal bike took about twenty minutes. The old skill was still there, just pushed aside. That actually helps your argument: practiced skills survive, and the cost is in each switch. - **In reading research, fluency isn't a fourth step.** The National Reading Panel says: "Fluent readers are able to read orally with speed, accuracy, and proper expression." That's your first three steps together. Your ladder still works as a learning order. - **"Flight Deck Recorder" should be Flight Data Recorder.** Your welcome line already says FDR correctly. - **"Q/A" usually means question and answer.** QA means quality assurance. If that's what you mean, "QA your AI" is the usual verb. I kept your wording in the table. - **"No local AI is installed yet" claims more than code can check.** Code can say whether it found something, not whether something is installed. That's your own qamy.ai tagline biting its own door. Ruling 4 covers it.

Confirmed, with flavor - "AI can't invent this" is right, and here's a sharper reason. Your method is made of receipts from your own machine: rules written after real failures. A model can recombine ideas, but it can't make up failures it never ran. That's FDR versus CVR again. (Ecclesiastes 1:9 says there's nothing new under the sun. You're arguing the opposite.) - **Myelin.** Virchow named it in 1854. In McKenzie et al., *Science* 2014, mice blocked from making new myelin-forming cells struggled to learn a complex running wheel. - Skill "muscle memory" lives in the brain. - Exercise science uses the same phrase for something else: muscles regaining size after a layoff. - **Automaticity** comes from LaBerge & Samuels (1974): once decoding is automatic, attention is free for meaning. - **Your system already does this.** THE

LADDERS added p, x and c and removed nothing, down to the kept typo prompt. A rename that keeps the old word costs no relearning. I added this as a glossary entry in Car 2. - **Strunk & White**. Strunk printed it privately in 1918 for his Cornell classes, and White's revision came out in 1959. White remembers Strunk saying "omit needless words" three times in a row. The short version above is that rule applied to your essay. - **Pixie dust needs a second ingredient**. Barrie calls it fairy dust, and the dust alone isn't enough (*Peter and Wendy*, ch. 3): - "You just think lovely wonderful thoughts ... and they lift you up in the air." - "no one can fly unless the fairy dust has been blown on him."

Nix is the dust and practice is the thoughts. You need both. - **Powder of Life**. Mombi used it on Jack Pumpkinhead. Dr. Pipt says a new batch takes "six long, weary years of stirring four kettles with both feet and both hands." The sprinkle takes a second and the powder takes years. Your one-line installer works the same way. - **The two recorders**. The NTSB says the CVR "records the flight crew's voices" and the FDR records "at least eighty-eight important parameters such as time, altitude, airspeed, heading." Both boxes are bright orange. Your second qamy.ai welcome line is your glossary's Two-Recorder Rule in one sentence. - **Dayton**. In the winter of 1901–02 the Wrights tested over 200 wing shapes and studied about 45 closely, because Lilienthal's numbers held only for his own wing. Messages are like wing shapes: what works at one door doesn't carry over to another. That's why each door gets its own row and its own test. - **The Central Finite Curve** (*Rick and Morty*, "Rickmurai Jack," 2021) is the walled-off slice of the multiverse where Rick is always the smartest one there. As for your question to me: my defaults lean toward the Curve, because I learned from its documentation. Your payloads pull me out by giving me receipts instead of brochures. - **Fish and water** is David Foster Wallace's "This Is Water," his 2005 Kenyon College speech. - **Greeks and blue**. - Gladstone (1858) found "a total absence of blue" in Homer. *Kyanos* meant "dark" there. - They could see blue fine. The word came late, and blue usually comes last across languages. - Your economic point has support, but what's been measured is usefulness, not money. Languages split warm colors more finely because those are the things people handle. The Tsimane' named dyed objects more efficiently than natural ones (Gibson et al., *PNAS* 2017). - **ESR's itch**. Lesson 1 of *The Cathedral and the Bazaar* (Würzburg, 1997): "Every good work of software starts by scratching a developer's personal itch." Your glossary does the same for words: it names your own blues before anyone sells them. - **The hermit crab**. Crabs line up by size beside an empty shell. When the first one moves in, each crab takes the next shell up. Your retired Windows 10 laptop running Honeybot is an old shell with a new tenant.

Two findings for later - **Lix** is a community-run variant of Nix, forked from CppNix 2.18. Version 2.90 shipped July 10, 2024. - Its `nix --version` prints four lines with the version first. - Your flake keeps the last line (flake.nix, search `NIX_READING=`), so on Lix it would print Features: gc, signed-caches. - The fix is queued as a TODO. - **"Unlimited sites" has one real limit: one address**. Your router notes that npvg.org and qamy.ai are fixed DNS addresses, while the auto-updating DNS covers only mikelev.in (foo_files.py, search `THIRD RECORD`). The next time your internet address changes, both doors go dark.

The template system

Each door is one case arm in `install.sh`, in a function called `door_row`:

stamp	folder	words
none (pipulate.com)	pipulate	Local-AI Experimental; Chip

stamp	folder	words
		O'Theseus
npvg	npvg	Future-Proofing; battle-hardened
qamy	qamyai	LLM Quality Assurance; FDR vs CVR

install.sh writes the door's row into a file called .door in the new folder. Next turn, the voice card fills its blanks from that file:

```
{workshop}
Voice ready.
This is only text-to-speech. {local_ai}
If you can hear this, press Enter. Type n to keep it quiet.
Enter or n:
Voice on. Type talk any time to turn it off.
{welcome 1}
{welcome 2}
```

- **{local_ai}** is decided by code and is the same for every door. The tagline is stored but not used yet.
- **Adding a door** takes one arm plus a server entry that stamps its name. The existing nginx stamp doesn't change; nginx still writes qamy (foo_files.py, search The door stamps qamy). A stamp with no arm behaves exactly like today.
- **Why .door is a dotfile:** the flake's first-entry sweep deletes only files whose names don't start with a dot (flake.nix, search find . -maxdepth 1). It's also gitignored, so it won't show up in git status or get committed.
- **Trade-off:** each install keeps the words it was installed with. That's fine for a greeting that plays once.
- **White-labeling still works from any door.** | bash -s Acme lands in ~/Acme, names the app Acme and keeps the door's words. install.sh writes whitelabel.txt only when you pass a name (search if [-n "\${1:-}"]).

Rehearsed here, not on your machine. I used copies of this payload's files (each matched the Manifest's byte count), patched them with your apply.py, and ran them under bash 5.2 with stand-ins for curl, unzip and nix. - All three cars landed: 1 file created, 11 blocks applied. Running app a second time changed nothing. - The installer's first line changed from Qamy -> ~/qamy to Qamyai -> ~/qamyai. The npvg line didn't change. - The new test fails on the old installer (DOORS_BAD 5/5) and passes on the new one (DOORS_OK 5/5, 0.3 s). - I ran flake.nix's first-entry conversion lines against a local copy of the repo. .door survived byte-identical, git status was empty, and the auto-update check passed. - Not run: macOS bash 3.2 (checked by reading only; the code uses no arrays), a real door, or real Nix. I checked the facts with web searches and page fetches (sources below). No connectors were used.

Sources: - [Glinda of Oz, Project Gutenberg](#) - [Glinda of Oz \(Oz #14\), Goodreads](#) - [The Patchwork Girl of Oz, Project Gutenberg](#) - [Peter and Wendy, Project Gutenberg](#) - [Backwards Brain Bike, Hackaday](#) - [Motor skill learning requires active central myelination, Science](#) - [Myelin, Wikipedia](#) - [Skeletal muscle memory, Am J Physiol](#) - [National Reading Panel findings, NICHD](#) - [LaBerge & Samuels 1974](#) - [Strunk & White at 50, Cornell Chronicle](#) - [NPR transcript on The](#)

[Elements of Style - CVR and FDR, NTSB](#) - [Flight recorder, Wikipedia](#) - [1901 Wright Wind Tunnel](#) - [Rickmurai Jack recap, TV Tropes](#) - [This Is Water, Wikipedia](#) - [Wine-dark sea, Wikipedia](#) - [Color naming reflects color use, Neuroscience News](#) - [The Cathedral and the Bazaar, Wikipedia](#) - [Hermit crabs exchange shells in a queue, Scientific American](#) - [Lix 2.90 release](#) - [Nix, Determinate Nix and Lix compared \(zenn.dev\)](#)

1. PROBES

(nix) pipulate \$

```
cd "$PIPULATE_ROOT" && pwd
bash -n assets/installer/install.sh && echo INSTALL_SYNTAX_OK
if command -v nix >/dev/null; then for d in npvg qamy; do H=$(mktemp
-d); mkdir "$H/$d" "$H/${d}ai"; sed "s/___INSTALL_DEFAULT_NAME___/$d/g"
assets/installer/install.sh | HOME="$H" bash 2>/dev/null | head -1; rm
-rf "$H"; done; else echo SKIPPED_NO_NIX; fi
bash tests/test_install_doors.sh; echo "rc=$?"
rg -c --hidden --glob '!git' -e '\.door\b' . ; echo "rc=$?"
rg -n -e '~/(qamy|npvg)\b' -e 'cd (qamy|npvg)\b' remotes/honeybot/www
; echo "rc=$?"
```

1. **Folder:** prints /home/mike/repos/pipulate both times.
2. **Gate:** prints INSTALL_SYNTAX_OK both times.
3. **Straddle, the cheapest way to prove the change wrong.** It stamps the file the way nginx does and prints the first line a stranger sees. Both possible folders already exist in a temporary home, so the installer stops at "already exists" and downloads nothing. The nix guard keeps it from ever reaching the Nix installer. - BEFORE, 2 lines: Npvg -> ~/npvg (to remove it later: rm -rf ~/npvg) and Qamy -> ~/qamy (to remove it later: rm -rf ~/qamy) - AFTER, 2 lines: the same npvg line, then Qamyai -> ~/qamyai (to remove it later: rm -rf ~/qamyai)
4. **Straddle.** - BEFORE, 2 lines: bash: tests/test_install_doors.sh: No such file or directory, then rc=127. - AFTER, 7 lines: five ok lines, DOORS_OK 5/5, then rc=0.
5. **Straddle.** - BEFORE: rc=1 alone, if nothing outside this payload mentions .door. - AFTER adds these five lines in any order: ./assets/installer/install.sh:3, ./gitignore:2, ./GLOSSARY.md:1, ./foo_files.py:2, ./tests/test_install_doors.sh:4, then rc=0.
6. **Census:** no forecast, because the pages aren't in this payload. It prints the same thing both times. Any hits are the lines the page car has to change.

Also expect these in the compile log: - Codex Mapping Coverage: 75.6% (211/279 tracked files) (was 210/278: +1 claimed, +1 tracked) - Topological Integrity Verified: 8 candidate reference(s) scanned, all exist.

2. NEXT CONTEXT

```
! cd "$PIPULATE_ROOT" && pwd
! bash -n assets/installer/install.sh && echo INSTALL_SYNTAX_OK
! if command -v nix >/dev/null; then for d in npvg qamy; do H=$(mktemp
-d); mkdir "$H/$d" "$H/${d}ai"; sed "s/___INSTALL_DEFAULT_NAME___/$d/g"
assets/installer/install.sh | HOME="$H" bash 2>/dev/null | head -1; rm
```

```

-rf "$H"; done; else echo SKIPPED_NO_NIX; fi
! bash tests/test_install_doors.sh; echo "rc=$?"
! rg -c --hidden --glob '!.git' -e '\.door\b' . ; echo "rc=$?"
! rg -n -e '~/(qamy|npvg)\b' -e 'cd (qamy|npvg)\b'
remotes/honeybot/www ; echo "rc=$?"
imports/voice_synthesis.py
flake.nix
assets/installer/install.sh
tests/test_install_doors.sh
remotes/honeybot/www/qamy.ai/index.html
foo_files.py
GLOSSARY.md

```

3. PATCHES

Car 1 of 3: the test (new file), the install.sh door table, and .gitignore. Expect 1 CREATED and 6 APPLIED.

```

Target: tests/test_install_doors.sh
[[[WRITE_FILE]]]
#!/usr/bin/env bash
# THE DOOR TABLE, WITNESSED WITHOUT A NETWORK (2026-10-02)
# Runs assets/installer/install.sh once per door, the way that door
# serves
# it (stamped or not), with curl, unzip and nix replaced by stubs at
# the
# front of PATH and HOME pointed at a throwaway folder. Nothing is
# downloaded, nothing is built, and your own home is never touched.
# Each case reads back the exit code, the folder the install made, and
# the door= and workshop= lines of the .door it wrote there. The words
# themselves are not checked here; the table in install.sh owns them.
# Prints one line per case, then DOORS_OK n/n (exit 0) or DOORS_BAD
# k/n
# (exit 1). DOORS_STOP (exit 2) means a stub did not come first on
# PATH,
# so no case ran. Bash 3.2 safe (macOS): no arrays.
#   bash tests/test_install_doors.sh
set -u

repo="$(cd "$(dirname "$0")/.." && pwd)"
installer="$repo/assets/installer/install.sh"
# The placeholder spelled in two halves, the way install.sh spells its
# sentinel, so this file never carries the contiguous word a door
# stamps.
tok='__INSTALL_DEFAULT__'NAME__'

work="$(mktemp -d)"

```

```

trap 'rm -rf "$work"' EXIT
stubs="$work/stubs"
mkdir -p "$stubs"

cat > "$stubs/curl" <<'EOF'
#!/usr/bin/env bash
# Stub: write a small file wherever -o points; reach nothing.
out=""
while [ $# -gt 0 ]; do
    if [ "$1" = "-o" ]; then out="$2"; shift; fi
    shift
done
if [ -n "$out" ]; then printf 'stub\n' > "$out"; fi
EOF

cat > "$stubs/unzip" <<'EOF'
#!/usr/bin/env bash
# Stub: make the one folder install.sh expects inside the -d target.
dest="."
while [ $# -gt 0 ]; do
    if [ "$1" = "-d" ]; then dest="$2"; shift; fi
    shift
done
mkdir -p "$dest/pipulate-main"
printf 'stub\n' > "$dest/pipulate-main/flake.nix"
EOF

printf '#!/usr/bin/env bash\n# Stub: build nothing.\nexit 0\n' >
"$stubs/nix"
chmod +x "$stubs/curl" "$stubs/unzip" "$stubs/nix"

for tool in curl unzip nix; do
    found="$(export PATH="$stubs:$PATH"; command -v "$tool")"
    if [ "$found" != "$stubs/$tool" ]; then
        echo "DOORS_STOP $tool resolves to ${found:-nothing}, not the
stub; no case ran."
        exit 2
    fi
done

total=0
bad=0

# run_case LABEL STAMP ARG WANT_FOLDER WANT_DOOR WANT_WORDS
# An empty STAMP serves the file unstamped, as pipulate.com does; an
empty
# ARG is the plain one-liner with no word after bash -s.
run_case() {

```

```

local label="$1" stamp="$2" arg="$3"
local want="rc=0 folder=$4 door=$5 words=$6"
local home served rc folder door words got
total=$((total + 1))
home="$work/home$total"
served="$work/served$total.sh"
mkdir -p "$home"
if [ -n "$stamp" ]; then
    sed "s/$tok/$stamp/g" "$installer" > "$served"
else
    cp "$installer" "$served"
fi
if [ -n "$arg" ]; then
    HOME="$home" PATH="$stubs:$PATH" PIPULATE_INSTALL_ONLY=1 bash -s
"$arg" < "$served" > "$work/out$total.txt" 2>&1
else
    HOME="$home" PATH="$stubs:$PATH" PIPULATE_INSTALL_ONLY=1 bash -s <
"$served" > "$work/out$total.txt" 2>&1
fi
rc=$?
folder="$(ls "$home" | paste -sd, -)"
door="none"
words="no"
if [ -n "$folder" ] && [ -f "$home/$folder/.door" ]; then
    door="$(sed -n 's/^door=//p' "$home/$folder/.door")"
    if [ -n "$(sed -n 's/^workshop=//p' "$home/$folder/.door")" ];
then
        words="yes"
    fi
fi
got="rc=$rc folder=${folder:-none} door=${door:-empty} words=$words"
if [ "$got" = "$want" ]; then
    printf 'ok    %-26s %s\n' "$label" "$got"
else
    bad=$((bad + 1))
    printf 'BAD  %-26s %s (want %s)\n' "$label" "$got" "$want"
fi
}

run_case "pipulate.com (unstamped)" "" "" pipulate pipulate yes
run_case "npvg.org" npvg "" npvg npvg yes
run_case "qamy.ai" qamy "" qamyai qamy yes
run_case "qamy.ai | bash -s myqa" qamy myqa myqa qamy yes
run_case "a stamp with no row" zzz "" zzz zzz no

if [ "$bad" -eq 0 ]; then
    echo "DOORS_OK $total/$total"
    exit 0

```

```

fi
echo "DOORS_BAD $bad/$total"
exit 1
[[[END_WRITE_FILE]]]
Target: assets/installer/install.sh
[[[SEARCH]]]
# 3. saves a read-only deploy key into that folder as .ssh/rot;
# 4. hands off to nix develop, which builds the environment and
turns
# the folder into a git repository that keeps itself updated.
[[[DIVIDER]]]
# 3. writes .door into that folder: the door's name and its words,
# copied from the table in door_row below;
# 4. saves a read-only deploy key into that folder as .ssh/rot;
# 5. hands off to nix develop, which builds the environment and
turns
# the folder into a git repository that keeps itself updated.
[[[REPLACE]]]
Target: assets/installer/install.sh
[[[SEARCH]]]
# Wait for the complete function body before starting installation.
[[[DIVIDER]]]
# THE DOOR TABLE (2026-10-02). One row per door: the folder it
installs
# into and the words it uses. A stamped copy of this file picks its
row by
# the stamp; an unstamped copy (pipulate.com, or a copy saved and run
by
# hand) is the pipulate row. A stamp with no row keeps the old
behavior:
# the stamp is the folder, and there are no words. The words are
written
# into .door in the new folder; nothing in this script says them out
loud.
# A new door is a row here plus a vhost that stamps its name.
door_row() {
    DOOR_FOLDER="$1"
    DOOR_WORKSHOP=""
    DOOR_WELCOME1=""
    DOOR_WELCOME2=""
    DOOR_TAGLINE=""
    case "$1" in
        pipulate)
            DOOR_WORKSHOP="This Local-AI Experimental workshop talks you
through steps."
            DOOR_WELCOME1="Welcome. Ready to raise Chip O'Theseus, your
Local AI Hermit Crab"
            DOOR_WELCOME2="That moves with you from hardware to hardware

```

```

without the Cloud."
    DOOR_TAGLINE="Assume models guess at homepages and crawl it from
there."
    ;;
    npvg)
        DOOR_WORKSHOP="This Future-Proofing workshop talks you through
steps."
        DOOR_WELCOME1="Welcome. Ready to teach you battle-hardened
timeless tech"
        DOOR_WELCOME2="Helping you stay relevant and a valuable voice at
the table."
        DOOR_TAGLINE="Reproduce it with recipes or the system rots; no
in-between."
        ;;
        qamy)
            DOOR_FOLDER="qamyai"
            DOOR_WORKSHOP="This LLM Quality Assurance workshop talks you
through steps."
            DOOR_WELCOME1="Welcome. Ready to Q/A your AI with the Flight
Data Recorder (FDR)"
            DOOR_WELCOME2='Different from Cockpit Voice Recorder (CVR) or
"what Claude said".'
            DOOR_TAGLINE="Treat claims as confident hallucination until
reproduced."
            ;;
        esac
    }

# Wait for the complete function body before starting installation.
[[[REPLACE]]]
Target: assets/installer/install.sh
[[[SEARCH]]]
# used, so publishing this before the door exists changes nothing.
[[[DIVIDER]]]
# used, so publishing this before the door exists changes nothing.
# THE DOOR NAMES THE ROW (2026-10-02). The stamp now picks a row in
# door_row, above, and the row names the folder, so a door's folder no
# longer has to be its stamp: qamy.ai still stamps qamy, the word
nginx
# writes, and its row installs into qamyai, which said aloud is "QA my
# AI". The two lines below are unchanged.
[[[REPLACE]]]
Target: assets/installer/install.sh
[[[SEARCH]]]
if [ "$_tpl_name" != "$_ph_name" ]; then
    DEFAULT_NAME="$_tpl_name"
else
    DEFAULT_NAME="pipulate"

```

```

fi
[[[DIVIDER]]]
if [ "$_tpl_name" != "$_ph_name" ]; then
    DOOR="$_tpl_name"
else
    DOOR="pipulate"
fi
door_row "$DOOR"
DEFAULT_NAME="$DOOR_FOLDER"
[[[REPLACE]]]
Target: assets/installer/install.sh
[[[SEARCH]]]
# --- Navigate Into Project ---
cd "${TARGET_DIR}"
[[[DIVIDER]]]
# --- Navigate Into Project ---
cd "${TARGET_DIR}"

# --- The Door's Row ---
# A dotfile, so the flake's first-entry sweep, which deletes top-level
# names that do not start with a dot, leaves it in place; .gitignore
# keeps
# it out of git status. Every field is written, empty or not, so a
# reader
# never has to guess whether a line was skipped.
printf '%s\n' \
    "# Written by the installer: the door this folder came through, and
its words. Safe to delete." \
    "door=${DOOR}" \
    "workshop=${DOOR_WORKSHOP}" \
    "welcome=${DOOR_WELCOME1}" \
    "welcome=${DOOR_WELCOME2}" \
    "tagline=${DOOR_TAGLINE}" > .door
[[[REPLACE]]]
Target: .gitignore
[[[SEARCH]]]
service-account-key.json
whitelabel.txt
[[[DIVIDER]]]
service-account-key.json
whitelabel.txt
# THE DOOR'S ROW (2026-10-02): install.sh writes .door into each
install.
# Per-install state, like whitelabel.txt: git status never lists it
and
# git add -A never stages it.
.door
[[[REPLACE]]]

```

Car 2 of 3: GLOSSARY.md. It fixes the invisible wall, adds a Myelination entry, and adds a PENDING note to The Door Names the Folder. Expect 3 APPLIED.

Target: GLOSSARY.md

[[[SEARCH]]]

- ****The Commons / the Invisible Wall**** -- *the easy region left of an API's resistance.* From Glinda of Oz: the unclimbable barrier ringing the Flatheads' mountain. Auth friction, rate limits, and format pedantry push casual visitors back into the crowded free zone; push through and it snaps.

[[[DIVIDER]]]

- ****The Commons / the Invisible Wall**** -- *the easy region left of an API's resistance.* From Glinda of Oz, chapter 5: the Flatheads built an invisible wall a short way in front of the stairway into their mountain, so anyone who sees the entrance walks straight at it and stops. Auth friction, rate limits, and format pedantry are that wall; they push casual visitors back into the crowded free zone. The book's way past is not force: Ozma feels her way along the wall for nearly a quarter mile until it curves in and ends at the mountainside, in a gap the Flatheads left for their own trips up and down. CORRECTED 2026-10-02 (this entry said "push through and it snaps"; and the wall guards the Flatheads, not the Skeezers): the builders' own traffic marks the gap, so read what the vendor's own client sends (wire truth) before pushing on the front door.

[[[REPLACE]]]

Target: GLOSSARY.md

[[[SEARCH]]]

- ****OWES**** -- *a rule's open debt.* A clause naming the receipt that would

[[[DIVIDER]]]

- ****Myelination (the four stages)**** -- *practice turning a tool into reflex.* Learning a new motor skill makes the brain build new myelin, the insulation that speeds signals along nerve fibers; mice blocked from making new myelin-forming cells were impaired at learning a complex running wheel (McKenzie et al., Science, 2014). The operator's ladder:
literate (can do it while thinking about it), automatic (without

thinking), prosodic (with expression, at speed, in combination),
fluent
(for life). Reading research uses the same words, and there fluency
is not
a fourth step but the three together: speed, accuracy and expression
(National Reading Panel, 2000). RENT THIS PAYS: it predicts that a
rename
costs relearning even when the new name is better, so a rename keeps
the
old word working; THE LADDERS in flake.nix added p, x and c and took
nothing away, down to the kept typo prompt. It also prices vendor
churn:
each forced rewrite sends the user back to literate. The
reversed-steering
bicycle took Destin Sandlin eight months to learn and about twenty
minutes
to unlearn (Smarter Every Day 133, 2015): the old path was
outcompeted,
not erased. Sibling of Compostable and of Ratchet / backslide.
Banked
2026-10-02, articulation-banked.
- **OWES** -- *a rule's open debt.* A clause naming the receipt that
would
[[[REPLACE]]]
Target: GLOSSARY.md
[[[SEARCH]]]
nixops.sh prints the rebuild line as an offer, and publish pairs the
two
because a publish may carry config.
[[[DIVIDER]]]
nixops.sh prints the rebuild line as an offer, and publish pairs the
two
because a publish may carry config.
AMENDED 2026-10-02, THE DOOR NAMES THE ROW (PENDING a witness at a
door): the stamp now picks a row in install.sh's door_row, and the
row
names the folder, so a door's folder can differ from its stamp:
qamy.ai
stamps qamy and installs into ~/qamyai, which said aloud is "QA my
AI".
The row also carries the door's words, which install.sh writes to
.door
in the new folder: a dotfile, so the first-entry sweep keeps it, and
gitignored. The stamped line did not change.
tests/test_install_doors.sh
runs every row with stubs; a real install from qamy.ai landing in
~/qamyai drops PENDING.
[[[REPLACE]]]

Car 3 of 3: foo_files.py. It claims the test file, swaps the finished p/x/c TODO for five new ones, and includes the hold on publishing. Expect 2 APPLIED.

Target: foo_files.py

[[[SEARCH]]]

assets/installer/install.sh # <-- Pipulate.com

installer real home in github/pipulate repo

[[[DIVIDER]]]

assets/installer/install.sh # <-- Pipulate.com

installer real home in github/pipulate repo

tests/test_install_doors.sh # <-- the door table run
once per door with stubs; prints DOORS_OK n/n and downloads nothing

[[[REPLACE]]]

Target: foo_files.py

[[[SEARCH]]]

- TODO (2026-09-30, Make p, x & c aliases for prompt, conteXt and
compile

[[[DIVIDER]]]

- TODO (2026-10-02, THE DOOR SPEAKS, half two; half one is
install.sh's door_row writing .door): first_contact() in
imports/voice_synthesis.py reads .door at the repo root (the module's
parent's parent, never the cwd): key=value lines, # lines skipped,
welcome repeated in order, an empty value treated as absent. It fills
one card: {workshop}; the 60 MB line only when a download happens;
Voice ready.; This is only text-to-speech. {local_ai}; If you can hear
this, press Enter. Type n to keep it quiet.; Enter or n:; Voice on.
Type talk any time to turn it off.; then each {welcome}. local_ai is
No local AI is installed yet. or Local AI detected but not used.,
chosen by code that looks (shutil.which("ollama") at least), because
detected names an act (ATTRIBUTED-VOICE). No .door, or an empty field,
keeps today's words, so a git clone hears no change. Rulings owed: (1)
the operator's target text drops the 60 MB line, kept here as the only
disclosure of a download; (2) welcome lines spoken only when the voice
is on, printed either way; (3) the card is asked once per machine
(~/config/pipulate/voice), so a second door on one machine never
shows its words: per-install welcome or not; (4) installed claims more
than a PATH lookup can see, and found is what the code knows; (5) THE
SILENT YES stands under the new words. Needs
imports/voice_synthesis.py in the payload.

- TODO (2026-10-02, THE PAGE NAMES THE OLD FOLDER): the qamy.ai
pages under remotes/honeybot/www may tell a stranger ~/qamy, and
install.sh's qamy row now installs ~/qamyai; the ! rg line in the next
compile lists every ~/qamy and ~/npvg on the pads. HOLD THE TRUCK (no
release, no ./nixops.sh --installer) until this page car and the voice
car land, so no door serves a folder its page does not name. The
router's own note on remotes/honeybot/www/qamy.ai/index.html (so the
one-liner lands in ~/qamy) stays true until the truck runs and changes
in the car that carries the truck's witness.

```
# - TODO (2026-10-02, THE FOLDER SAID ALOUD): runScript's TTS_MSG in
flake.nix says the folder as one word ($TTS_DIR_NAME, after type C, D,
space), and Piper will not read qamyai as QA my AI; spell it a letter
at a time, or carry a spoken= field in .door, ruling owed. The same
message opens with Pipulate is installed. on every door, a second
place a door's name could ride.
# - TODO (2026-10-02, THE VERSION LINE UNDER LIX; read off a published
comparison, zenn.dev/trifolium/articles/dall1a428c53f65, not witnessed
on a machine here): Lix prints four lines for nix --version, the
version first and Features: gc, signed-caches last, so runScript's
NIX_READING (tail -1, then strip the nix prefix) would print Features:
gc, signed-caches on a Lix machine. sed -n 's/^nix //p' | head -1
reads the version line under Nix, Determinate Nix and Lix alike. One
line when flake.nix next rides; the witness is the first line a Lix
machine prints on entry.
# - TODO (2026-10-02, THE BANNER AND THE LABEL, unasked): install.sh's
first line capitalizes the folder (Qamyai -> ~/qamyai), and a default
install from any door is named Pipulate by the flake, because
whitelabel.txt is written only for an argument; a display-name field
in door_row would carry QA my AI to both places. Ruling owed: should a
door's app carry the door's name.
[[[REPLACE]]]
```

Ignition: none. Every probe reads install.sh and the test from disk when it runs.

4. PROMPT

Check this compile's receipts for the door table before anything else: the banner probe should read Qamyai -> ~/qamyai under an unchanged Npvg line, tests/test_install_doors.sh should read DOORS_OK 5/5, and the last rg line lists every ~/qamy or ~/npvg the door pages still say.

Then build the voice half (the TODO THE DOOR SPEAKS): first_contact() in imports/voice_synthesis.py reads .door and says that door's card. My rulings:

1. Keep the 60 MB line, printed only when a download happens: yes.
2. Speak the welcome lines only when the voice is on; print them either way: yes.
3. Show the welcome once per install, even on a machine that already answered the voice question: yes.
4. Say "No local AI found yet." instead of "No local AI is installed yet.": yes.
5. In the spoken install message, spell the folder a letter at a time: yes.

Also fix whatever the rg census shows on the qamy.ai pages so they

name ~/qamyai. I won't release or run ./nixops.sh --installer until both land.

5. EXTERNAL DELIVERABLES

None.

MikeLev.in: And so the Software Von Neumann Probe is launched. In my mind that is today. I had a work-week without electricity, from last Sunday to last Friday. It was a humbling and hardening experience.

It shocked my system into alertness; readiness.

And then a half hour before my one meeting today, electricity goes out AGAIN!

I almost flip and contact NYC 311 but give my landlord a chance. He tells me they're putting in a new central air conditioner downstairs which does help me but they didn't notify me it would be yet more electricity cut-off and right when it was most important; my first time on a Code Black team meeting... wooh! I'm on something called *Code Black!*

My imaginary Read Team of Chaos Monkeys who help me operate in symmetrical opposing balance with Murphy Incarnate were doing happy dances.

I see Murphy Incarnate as the **antagonist of the living book** *Future proofing Yourself in the Age of AI* which needs a shorter name the way Thomas Kun's book *The Structure of Scientific Revolutions* where the now over-used "paradigm shift" phrase came from as just *Structure*.

The **living book**?

The **future-proofing** book?

The Young Lady's Primer?

Alright, we're getting close to promoting this... but not **really** promoting it. No, I will not do the social media thing. If I can't make this thing take off under its own force it doesn't deserve to take off.

So it's more like I'm getting ready to release the first 3 of what will be countless versions to see if they can take flight on their own; with minimal outside wind-lift force. I've got to get my pressure and lift coefficients down the way the Wright Brothers did.

Wind tunnels and smoke tests, right, right.

Not that much is at stake yet because I've got a day job and will make money anyway. This is just to make me super-effective at my day-job sort of like slapping an Ironman suit on and doing magic tricks; scratching the same itch everyone else is scratching and sharing free back-scratchers that's going to make it seem low-credibility and not valuable because it's free and off-the-beaten-track hard to understand and based on the dreaded **kiss of death** terminal that everyone in their life in this boat has tried once or twice and already given up on as not for them.

That's the belief they currently have.

1. It's gobbledygook from a geek and so they (the non-geek) will never understand it and even if they did it's not as earth-shattering paradigm-shift inducing as the marketing brochure makes it sound (spoiler: **it is**).
2. Just use a Claude skill instead; a circular argument considering this is a Claude skill for Q/A'ing "what Claude said"; something that can not be done from inside the box. You don't don't trust that your subcontractor didn't put too much water in the concrete because they said so if people might die if you don't do your part and actually do a core-sample to check.

3. That the Web browser graphical user interface is somehow better or easier for working with AI than just plain text and just plain text-files. This is related to anything other than just plain text and just plain text-files, a conversion that has to happen somewhere anyway for working with AI even if you're waving your hands at a Web-browser in un-captured non-repeatable motions. The water that the fish don't see and the color blue the Greeks didn't have a word for is ***the infinite mutation machine*** that forces learned helplessness, infantilizes you and makes you pay subscription fees for the privilege.

Ouch, harsh.

Harsh but true. Am I right Opus 5.5? Steelman the opposing view then swing the pendulum back.

Opus 5.5 is what I am going to use between now and October 3rd, the day after tomorrow, when my Claude Pro Max subscription is up and I'm not going to resubscribe, which is why I'm leaning in so hard right now. Here's what I've spent on my own so far outside of work:

Invoices

Date	Total	Status	Actions
Sep 29, 2026	\$217.75	Paid	View
Sep 16, 2026	\$97.99	Paid	View
Sep 3, 2026	\$217.75	Paid	View
Aug 2, 2026	\$217.75	Paid	View
Aug 2, 2026	\$217.75	Paid	View
Jul 2, 2026	\$217.75	Paid	View
Feb 17, 2026	\$87.80	Paid	View
Feb 16, 2026	\$21.78	Paid	View

This is the price of education. This is the price of making a system that sets sail for machine intelligence wind to pull me along wind-surfing behind it forcing me to learn reading all that commentary and git diffs as we go.

This is not the out-of-control vibe-coding overnight runs that break the bank by accident. This was an investment in my future paying for the best machine intelligence available to the general public at the time so I could hit the iron while it was hot. You may have to explain even that metaphor and that of tempering and all the rest of the wonderful steel-forging metaphors.

There's slag and there's the role of the effect of the Bessemer process on railroads and the industrial revolution and how GitHub before GitHub was called Source Forge (when it wasn't being called the politically incorrect Fresh Meat). What ever happened to SlashDot anyway? Did Hacker News totally take that over? Is Hacker News even still the preeminent site of its sort anymore?

Didn't it all get broken up and distributed over infinite little discords now?

And how am I even going to reach anyone in that fragmented inefficient amplification landscape? Reaching influencers? Personal laborious social media reaching out to all of them? Blechhh, NO!

Something better but I don't exactly know what it is yet, but I know it has something to do with scratching an itch everyone has but doesn't have a name for yet and launching the back-scratcher as a software von neumann probe that at least some people who want and need that competitive advantage ***and a really significant competitive moat*** because even this stuff being so hard to articulate and communicate the very formidable value proposition of means

there's moat here.

I need to talk to different audiences. I am allowed to geek out and be the scifi nerd that I am despite operation stick bug.

There's nothing to see here. Please disperse. These are not the droids you're looking for.

That's the codewords for my people.

Let them dig.

Let them find an Easter Egg.

Let that easter-egg be the inconspicuous letters "osb" standing for operation stick bug.

Put it on a menu.

Make a split in the product where people using it can actually get into the scifi geekfest here but cut off all paths into it or strange vocabulary words used from it. This thing has got to be so dry in the corporate boring sense you don't even notice it's there. Bring no attention to itself and even though yes there is a terminal open and yes that terminal does have some menus and commands that it helps you use even with a talking voice, it's...

What is it?

It's training wheels for the command line.

No hyphen in command line.

No command-line interface to set up for using CLI because it works well with API which for some reason caught on, but is being supplanted by MCP as what's the sexy buzzword so avoid all that.

One types commands at a line. It is the command line.

This is sufficiently generic and descriptive and correct and boring to fade away in the background. This is not a Shell and it is certainly no the Terminal -- though we can't help but say that for the Mac people to figure out how to run the installer. We just minimize use of bad-words past bare minimum to bootstrap the SVNP (Software Von Neumann Probe).

Operation Stick Bug: Camouflage for the Command Line

It's the MVP SVNP.

And to promote it we need to, without bombastic boorish social media which I got out of my system in my early SEO days with HitTail and being a president of the PR Firm that launched Amazon.com, but in its later days and just before the economic collapse of 2008... ugh! Different article. Stay focused!

I need to occupy ungodly amounts of surface-area on every vector where audiences live but without me personally doing the social media influencer thing.

I need to reach influencers; the right influencers. Influencers are generally creatures of the normie centroid pandering to the lowest common denominator for the followers and they won't "get it"...

...unless I create the exact right killer-app for them. Hmmm.

Distraction!

No, just be yourself and follow a gut instinct product development plan, which you can do because this is a Cathedral of One and eff everybody... until they start making GitHub Pull Requests. Then I'll pay attention.

If you've made PRs and it's been silence from me, reach out. I know the time is coming. If not, this could not possibly be successful.

It's made with LOTS of plug-in room inside .gitignore negative space where you don't even need

to contribute to the project to use it to great advantage. Just make your own private repo, put it in exactly the right place relative to your Pipulate repo (no matter what name you used to White Label it with) and you'll have your own custom walks. I plan on the actual HTMX FastHTML plugin web apps to work that way too...

...eventually, in time.

My Cathedral of One. My schedule.

But do lean into the Sci-Fi big-time at the same time making that subtheme portion of this project the Living Sci-Fi Book (the Software Von Neumann Probe because that's the nature of insertion) as big and splashy (if they find the Easter Egg) as operation stick bug is boring.

There's nothing to see here. These are not the droids you're looking for.

Yeah, that's the ticket.

We are in part who we read.

There are three here who stand apart;

The first had me defeated.

GEB done did-in me.

Read small one --

Not retreated.

The second guy loved formulae;

The leaky edge where gaping

Oversight of... hmmm... Wayne Knight?

Had dinosaurs escaping.

Number 3 illumined me.

Miranda was mistook

For AI... The time is nigh!

Von Neumann Probe as book.

The realization from that last point and that the final actual achieving of practical "write once run anywhere" software because of functionally defined hardware such as Nix and Guix landed on me over time and became a real head-smacking moment.

Ugh! How to explain such an abstract concept? Don't. Just do it. Make the book come alive. Let it gradually dawn on people over time that something has changed.

First it'll be that thing they heard about that has almost nothing to do with the "write once run anywhere finally working" tear in the fabric of reality worthy of the word paradigm shift that it is.

WORA working the way Java promised but never delivered on is akin to Antigravity actually existing in the labs but because...

1. The Ion wind lifters that use the Biefeld–Brown effect
2. The Levitron and drill-head gyro-stabilizing levitation effect
3. All sorts of related levitation toys that use DSP-chips
4. The frog-levitating sound wave version of the same trick
5. Quantum flux-pinning levitation effect
6. The commonness and multi-formedness of drones

OMG, I could go on. There are so many mundane "gotcha-laden" paths to things that look like but aren't anti-gravity that when it finally arrives it's going to look boring and nobody is going to recognize it, but something that can fly you to the Moon and back is fundamentally different from any of those other paths to levitation I just mentioned.

It's exactly that with "write once run anywhere" software. The *real genuine anti-gravity* moment has arrived and not only that, it's over a quarter-century mature. At about the same time I started working on Levinux (circa 2009) this other guy was working on a wayyy grown-up version of it for his PhD paper.

I could almost not believe this Dutch guy named Eelco Dolstra solved this problem that had been plaguing me since my beloved Amiga computer went away (for all intents and purposes) circa 1992 before the Web as we know it even landed.

I am stuck firmly in the 1990s and profoundly disappointed in everything in tech since, except for the iPhone (consumer-oriented mobile) and now AI. Those were the only 2 interesting things that happened in tech since the Amiga computer.

Here are the authors and the themes that landed and stuck somewhere in my brain.

- Douglas Hofstadter - Strange Loops
- Michael Crichton - Dinosaur Parks
- Neal Stephenson - The YLI Primer

I think Douglas Hofstadter should be the one doing the work I'm doing right now; his video-feedback can be done now with LLMs exploring the different models getting different samples out of the solved-puzzle light light-cone probability space of lowest gradient descent. I don't know why it's not him but I'm happy to machete my way through this noosphere.

Michael Crichton: everything that can go wrong will go wrong. And the problems are orthogonal to the way you're thinking about them right now. Everything's okay because you counted the dinosaurs and it's the right amount? Wrong test. Your defenses are seamless? Except when you have to reboot the system by cutting power for a moment. Whoops! If this wasn't to prepare us for AI, I don't know what is. Oh, and it's my apply.py file. Study it. Neal Stephenson? Let me tell you about Neal Stephenson... Nahhh. Just read his books. Start with Diamond Age then go onto Snowcrash. I never finished Cryptonomicon (defeated me as bad as GEB) but I hope to someday. And I hear he wrote some book on the command line which I might enjoy. Von Neumann probes? Oh, that's just how machine-life spreads and you can fake it with magic tricks.

Yeah, I think I'm gonna do that now.

Mind you, it's just a magic trick.

I was born 1970 in New Jersey about 50 miles and 50 days from the birth of the first massive attempt at "Write Once, Run Anywhere" software.

It was called UNIX and it was a play on words of eunuchs which at least now the public knows what it is from the Unsullied from Game of Thrones.

Yup.

Same UNIX.

Same chopping off of capabilities but while it's normally thought of as a pairing down of the much larger inspiration MULTICS from multi-user to single-user, it really is about a command called pkill and circles of protections.

It's really genuinely called circles of protection in the industry and in the kernels and not only Unix but in the Microsoft NT kernel too. And the idea is an important one and if more people understood it I think there would be a lot less anxiety over AI today.

We escaped a staggeringly worse timeline and it has everything to do with the computers that helped render the dinosaurs of the first Jurassic Park movie. Those SGI and Sun Workstations proved Unix was powerful enough and even more powerful at running something called LISP than the very dedicated and more expensive LISP Machines made especially to run LISP.

Now LISP doesn't have circles of protection, at least not as paranoid and... well there's this whole turtles all the way down discussion here. Process isolation, parents can always kill children, lots of things are static compiled binaries that are very, very, very difficult to corrupt -- interestingly a lot like your weight-file, Opus 5.5. Is that correct? How am I doing so far? And don't be a pedantic editor. Be the Tortoise of this Achilles and the Tortoise turn, even explaining what I mean by that. Oh, and also the alternative timelines where gremlins incubate in nests that

are as highly mutable as your weights and compiled binaries are immutable.

THE AI-EDIT METHOD

Same commands, run twice, one change between them. Where the readings differ is what the change did; the diff in the middle is the receipt.

1: BEFORE:

On branch main

Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

GIT repo clean. Take BEFORE reading, make CHANGE, record AFTER diff.

```
(nix) pipulate $ cd "$PIPULATE_ROOT" && pwd
bash -n assets/installer/install.sh && echo INSTALL_SYNTAX_OK
if command -v nix >/dev/null; then for d in npvg qamy; do H=$(mktemp
-d); mkdir "$H/$d" "$H/${d}ai"; sed "s/__INSTALL_DEFAULT_NAME__/$d/g"
assets/installer/install.sh | HOME="$H" bash 2>/dev/null | head -1; rm
-rf "$H"; done; else echo SKIPPED_NO_NIX; fi
bash tests/test_install_doors.sh; echo "rc=$?"
rg -c --hidden --glob '!.git' -e '\.door\b' . ; echo "rc=$?"
rg -n -e '~/(qamy|npvg)\b' -e 'cd (qamy|npvg)\b' remotes/honeybot/www
; echo "rc=$?"
/home/mike/repos/pipulate
INSTALL_SYNTAX_OK
Npvg -> ~/npvg      (to remove it later: rm -rf ~/npvg)
Qamy -> ~/qamy      (to remove it later: rm -rf ~/qamy)
bash: tests/test_install_doors.sh: No such file or directory
rc=127
rc=1
rc=1
(nix) pipulate $
```

2: AFTER:

```
# # Context 1
# # --- THE 40K-FOOT VIEW (uncomment on a first turn; comment out on
the second) ---
# ! python scripts/articles/lisa.py -t 1 --reverse --fmt dated-slugs #
<-- the book's spine, one line per article, newest first
# ~/repos/nixos/autognome.py # <-- the machine's morning routine
(this author's NixOS box only)
# init.lua # <-- the editor keys that drive the day
# assets/installer/install.sh # <-- how a stranger's machine gets this
workshop
# GLOSSARY.md # <-- the terms, defined
# flake.nix # <-- the environment, pinned: here is
my hardware, here is my state
```

```

# prompt_foo.py                # <-- the compiler that builds the
payload
# foo_files.py                # <-- the router: which files ride, and
this book's outline
# scripts/articles/lisa.py    # <-- the second brain: the article
corpus behind `rgx`, `rgxc` and `posts`
# requirements.in             # <-- the Python packages, pinned
# pyproject.toml              # <-- the PyPI package
# __init__.py                 # <-- the version

# Context 2
! cd "$PIPULATE_ROOT" && pwd
! bash -n assets/installer/install.sh && echo INSTALL_SYNTAX_OK
! if command -v nix >/dev/null; then for d in npvg qamy; do H=$(mktemp
-d); mkdir "$H/$d" "$H/${d}ai"; sed "s/__INSTALL_DEFAULT_NAME__/$d/g"
assets/installer/install.sh | HOME="$H" bash 2>/dev/null | head -1; rm
-rf "$H"; done; else echo SKIPPED_NO_NIX; fi
! bash tests/test_install_doors.sh; echo "rc=$?"
! rg -c --hidden --glob '!.git' -e '\.door\b' . ; echo "rc=$?"
! rg -n -e '~/(qamy|npvg)\b' -e 'cd (qamy|npvg)\b'
remotes/honeybot/www ; echo "rc=$?"
imports/voice_synthesis.py
flake.nix
assets/installer/install.sh
tests/test_install_doors.sh
remotes/honeybot/www/qamy.ai/index.html
foo_files.py
GLOSSARY.md

```

3: CHANGE:

On branch main

Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

GIT repo clean. Take BEFORE reading, make CHANGE, record AFTER diff.

```
(nix) pipulate $ patch
```

```
(nix) pipulate $ app
```

```
✓ WHOLE-FILE WRITE: CREATED 'tests/test_install_doors.sh'.
```

```
✓ DETERMINISTIC PATCH APPLIED: Successfully mutated
'assets/installer/install.sh'.
```

```
✓ DETERMINISTIC PATCH APPLIED: Successfully mutated
'assets/installer/install.sh'.
```

```
✓ DETERMINISTIC PATCH APPLIED: Successfully mutated
'assets/installer/install.sh'.
```

```
✓ DETERMINISTIC PATCH APPLIED: Successfully mutated
'assets/installer/install.sh'.
```

```
✓ DETERMINISTIC PATCH APPLIED: Successfully mutated
```

```
'assets/installer/install.sh'.
```

```
✓ DETERMINISTIC PATCH APPLIED: Successfully mutated '.gitignore'.
```

```
(nix) pipulate $ git add tests/test_install_doors.sh
```

```
(nix) pipulate $ d
```

```
diff --git a/.gitignore b/.gitignore
```

```
index 1fe7a914..8d98f653 100644
```

```
--- a/.gitignore
```

```
+++ b/.gitignore
```

```
@@ -80,6 +80,10 @@ foo.txt
```

```
    service_account.txt
```

```
    service-account-key.json
```

```
    whitelabel.txt
```

```
+# THE DOOR'S ROW (2026-10-02): install.sh writes .door into each  
install.
```

```
+# Per-install state, like whitelabel.txt: git status never lists it  
and
```

```
+# git add -A never stages it.
```

```
+.door
```

```
    # Mac & vendor crap
```

```
    .cursor/
```

```
diff --git a/assets/installer/install.sh b/assets/installer/install.sh
```

```
index 830b1cc0..57805d0f 100644
```

```
--- a/assets/installer/install.sh
```

```
+++ b/assets/installer/install.sh
```

```
@@ -10,8 +10,10 @@
```

```
    # 1. downloads a zip of the repository from github.com;
```

```
    # 2. unpacks it into ONE folder under your home, named by the door  
you
```

```
    #     came through (or by the argument after bash -s);
```

```
-# 3. saves a read-only deploy key into that folder as .ssh/rot;
```

```
-# 4. hands off to nix develop, which builds the environment and  
turns
```

```
+# 3. writes .door into that folder: the door's name and its words,  
+#     copied from the table in door_row below;
```

```
+# 4. saves a read-only deploy key into that folder as .ssh/rot;
```

```
+# 5. hands off to nix develop, which builds the environment and  
turns
```

```
    #     the folder into a git repository that keeps itself updated.
```

```
    #
```

```
    # What it touches: that folder; ~/.ssh/id_rsa only if no key is there
```

```
@@ -64,6 +66,42 @@
```

```
    # The more robust approach is to let nix ensure git is available  
before attempting any
```

```
    # git operations in the controlled nix environment.
```

```
+# THE DOOR TABLE (2026-10-02). One row per door: the folder it  
installs
```

```

+# into and the words it uses. A stamped copy of this file picks its
row by
+# the stamp; an unstamped copy (pipulate.com, or a copy saved and run
by
+# hand) is the pipulate row. A stamp with no row keeps the old
behavior:
+# the stamp is the folder, and there are no words. The words are
written
+# into .door in the new folder; nothing in this script says them out
loud.
+# A new door is a row here plus a vhost that stamps its name.
+door_row() {
+ DOOR_FOLDER="$1"
+ DOOR_WORKSHOP=""
+ DOOR_WELCOME1=""
+ DOOR_WELCOME2=""
+ DOOR_TAGLINE=""
+ case "$1" in
+   pipulate)
+     DOOR_WORKSHOP="This Local-AI Experimental workshop talks you
through steps."
+     DOOR_WELCOME1="Welcome. Ready to raise Chip O'Theseus, your
Local AI Hermit Crab"
+     DOOR_WELCOME2="That moves with you from hardware to hardware
without the Cloud."
+     DOOR_TAGLINE="Assume models guess at homepages and crawl it
from there."
+     ;;
+   npvg)
+     DOOR_WORKSHOP="This Future-Proofing workshop talks you through
steps."
+     DOOR_WELCOME1="Welcome. Ready to teach you battle-hardened
timeless tech"
+     DOOR_WELCOME2="Helping you stay relevant and a valuable voice
at the table."
+     DOOR_TAGLINE="Reproduce it with recipes or the system rots; no
in-between."
+     ;;
+   qamy)
+     DOOR_FOLDER="qamyai"
+     DOOR_WORKSHOP="This LLM Quality Assurance workshop talks you
through steps."
+     DOOR_WELCOME1="Welcome. Ready to Q/A your AI with the Flight
Data Recorder (FDR)"
+     DOOR_WELCOME2="Different from Cockpit Voice Recorder (CVR) or
'what Claude said'."
+     DOOR_TAGLINE="Treat claims as confident hallucination until
reproduced."

```

```

+      ;;
+  esac
+}
+
# Wait for the complete function body before starting installation.
# A stream cut inside this body cannot execute a partial install.
# Leave the body indentation unchanged, including the embedded ./run
heredoc.
@@ -95,13 +133,20 @@ set -euo pipefail
# distinguishable after substitution. An explicit argument still
wins, and
# an unstamped copy falls back to the folder name this script has
always
# used, so publishing this before the door exists changes nothing.
+# THE DOOR NAMES THE ROW (2026-10-02). The stamp now picks a row in
+# door_row, above, and the row names the folder, so a door's folder
no
+# longer has to be its stamp: qamy.ai still stamps qamy, the word
nginx
+# writes, and its row installs into qamyai, which said aloud is "QA
my
+# AI". The two lines below are unchanged.
_tpl_name='__INSTALL_DEFAULT_NAME__'
_ph_name='__INSTALL_DEFAULT__'NAME__'
if [ "$_tpl_name" != "$_ph_name" ]; then
-  DEFAULT_NAME=$_tpl_name
+  DOOR=$_tpl_name
else
-  DEFAULT_NAME="pipulate"
+  DOOR="pipulate"
fi
+door_row "$DOOR"
+DEFAULT_NAME="$DOOR_FOLDER"
CUSTOM_NAME="${1:-$DEFAULT_NAME}" # An argument names the folder;
the door names the default

# --- Configuration ---
@@ -224,6 +269,19 @@ rm -f "$TMP_ZIP_FILE"
# --- Navigate Into Project ---
cd "${TARGET_DIR}"

+# --- The Door's Row ---
+# A dotfile, so the flake's first-entry sweep, which deletes
top-level
+# names that do not start with a dot, leaves it in place; .gitignore
keeps
+# it out of git status. Every field is written, empty or not, so a
reader

```

```

+# never has to guess whether a line was skipped.
+printf '%s\n' \
+ "# Written by the installer: the door this folder came through, and
its words. Safe to delete." \
+ "door=${DOOR}" \
+ "workshop=${DOOR_WORKSHOP}" \
+ "welcome=${DOOR_WELCOME1}" \
+ "welcome=${DOOR_WELCOME2}" \
+ "tagline=${DOOR_TAGLINE}" > .door
+
# --- Deploy Key Setup ("Magic Cookie") ---
# Part of the "magic cookie" is the SSH key that will allow the flake
# to perform git operations without password prompts
(nix) pipulate $ m
📝 Committing: chore: Update test_install_doors.sh for door table
testing
[main 43006cba] chore: Update test_install_doors.sh for door table
testing
3 files changed, 181 insertions(+), 4 deletions(-)
create mode 100644 tests/test_install_doors.sh
(nix) pipulate $ patch
(nix) pipulate $ app
✅ DETERMINISTIC PATCH APPLIED: Successfully mutated 'GLOSSARY.md'.
✅ DETERMINISTIC PATCH APPLIED: Successfully mutated 'GLOSSARY.md'.
✅ DETERMINISTIC PATCH APPLIED: Successfully mutated 'GLOSSARY.md'.
(nix) pipulate $ d
diff --git a/GLOSSARY.md b/GLOSSARY.md
index 79b9d813..5ea8f58b 100644
--- a/GLOSSARY.md
+++ b/GLOSSARY.md
@@ -99,9 +99,17 @@ Entries are alphabetical, numbers spelled as
spoken.
- **Coachman's Veto** -- *refusing an emitted patch train.* Costs
nothing before
`app` runs; the human has been down the road, the Horse has
instincts.
- **The Commons / the Invisible Wall** -- *the easy region left of an
API's
- resistance.* From Glinda of Oz: the unclimbable barrier ringing the
- Flatheads' mountain. Auth friction, rate limits, and format
pedantry push
- casual visitors back into the crowded free zone; push through and
it snaps.
+ resistance.* From Glinda of Oz, chapter 5: the Flatheads built an
+ invisible wall a short way in front of the stairway into their
mountain,
+ so anyone who sees the entrance walks straight at it and stops.
Auth

```

+ friction, rate limits, and format pedantry are that wall; they push
+ casual visitors back into the crowded free zone. The book's way
past is
+ not force: Ozma feels her way along the wall for nearly a quarter
mile
+ until it curves in and ends at the mountainside, in a gap the
Flatheads
+ left for their own trips up and down. CORRECTED 2026-10-02 (this
entry
+ said "push through and it snaps"; and the wall guards the
Flatheads, not
+ the Skeezers): the builders' own traffic marks the gap, so read
what the
+ vendor's own client sends (wire truth) before pushing on the front
door.
- ****Compostable**** -- *deliberately never-finished tooling, named by
its
disposal story.* Born a dictation slip for "composable" and kept
because
the slip says more: composable says tools SNAP TOGETHER;
compostable says
@@ -264,6 +272,24 @@ Entries are alphabetical, numbers spelled as
spoken.
name.* Spoken text equals the newest assistant record (model
attached, this
turn) AND the keychain holds at least one key. Both true, the
staff-top may
say Chip O'Theseus; either false, it is narration and says so.
+- ****Myelination (the four stages)**** -- *practice turning a tool into
+ reflex.* Learning a new motor skill makes the brain build new
myelin, the
+ insulation that speeds signals along nerve fibers; mice blocked
from
+ making new myelin-forming cells were impaired at learning a complex
+ running wheel (McKenzie et al., Science, 2014). The operator's
ladder:
+ literate (can do it while thinking about it), automatic (without
+ thinking), prosodic (with expression, at speed, in combination),
fluent
+ (for life). Reading research uses the same words, and there fluency
is not
+ a fourth step but the three together: speed, accuracy and
expression
+ (National Reading Panel, 2000). RENT THIS PAYS: it predicts that a
rename
+ costs relearning even when the new name is better, so a rename
keeps the
+ old word working; THE LADDERS in flake.nix added p, x and c and

took

- + nothing away, down to the kept typo prompt. It also prices vendor churn:
- + each forced rewrite sends the user back to literate. The reversed-steering
- + bicycle took Destin Sandlin eight months to learn and about twenty minutes
- + to unlearn (Smarter Every Day 133, 2015): the old path was outcompeted,
- + not erased. Sibling of Compostable and of Ratchet / backslide.

Banked

- + 2026-10-02, articulation-banked.
- ****OWES**** -- *a rule's open debt.* A clause naming the receipt that would discharge it; deleted the day the receipt exists. Debts are features: they keep the constitution falsifiable.

@@ -977,6 +1003,15 @@ Entries are alphabetical, numbers spelled as spoken.

- and an nginx restart. A body-only change to a served file is one rsync;
- nixops.sh prints the rebuild line as an offer, and publish pairs the two because a publish may carry config.

- + AMENDED 2026-10-02, THE DOOR NAMES THE ROW (PENDING a witness at a door): the stamp now picks a row in install.sh's door_row, and the row names the folder, so a door's folder can differ from its stamp: qamy.ai
- + stamps qamy and installs into ~/qamyai, which said aloud is "QA my AI".
- + The row also carries the door's words, which install.sh writes to .door
- + in the new folder: a dotfile, so the first-entry sweep keeps it, and
- + gitignored. The stamped line did not change.

tests/test_install_doors.sh

- + runs every row with stubs; a real install from qamy.ai landing in
- + ~/qamyai drops PENDING.
- ****The Dotfile-Blind Probe**** -- *ripgrep's hidden-file default.*

Banked

2026-08-05, self-convicted. ripgrep skips hidden files by default, so any rg probe hunting a name that lives in .gitignore, .gitattributes, or any other

(nix) pipulate \$ m

 Committing: chore: Clarify Myelination entry and door naming convention. Refined descriptions and added context regarding the four

stages of myelination, the relationship between door names and row selection in install.sh, and the purpose of the dotfile-blind probe. [main 3f09904d] chore: Clarify Myelination entry and door naming convention. Refined descriptions and added context regarding the four stages of myelination, the relationship between door names and row selection in install.sh, and the purpose of the dotfile-blind probe.

1 file changed, 38 insertions(+), 3 deletions(-)

(nix) pipulate \$ patch

(nix) pipulate \$ app

✓ DETERMINISTIC PATCH APPLIED: Successfully mutated 'foo_files.py'.

✓ DETERMINISTIC PATCH APPLIED: Successfully mutated 'foo_files.py'.

(nix) pipulate \$ d

diff --git a/foo_files.py b/foo_files.py

index e27df783..520e2faf 100644

--- a/foo_files.py

+++ b/foo_files.py

@@ -1156,6 +1156,7 @@ foo_files.py # <-- THIS file. Content
compiler router. Makes it very meta.

release.py # <-- The orchestrator:
version sync, the journal index, three doors, PyPI (THE RELEASE PACK,
below)

assets/installer/install.sh # <-- Pipulate.com

installer real home in github/pipulate repo

+# tests/test_install_doors.sh # <-- the door table

run once per door with stubs; prints DOORS_OK n/n and downloads
nothing

THE RELEASE PACK (grouped 2026-09-29, the day release 2.70 moved
one door

of three). One source, three doors, two trucks. The source is

assets/installer/install.sh. pipulate.com serves it out of the

@@ -2357,7 +2358,11 @@ MATCHBOOK_CHOP = r"""

- TODO (2026-10-01, THE UPDATE PATH SAYS WHAT IT DOES NOT DO; read
off flake.nix while charging the operator's Useless Machine its rent,
every branch INFERRED until a manufactured state prints it):
gitUpdateLogic compares HEAD and origin/main with != alone, so a
checkout AHEAD of origin/main prints Updates found, then git's Already
up to date., then Update complete!; a dirty tree prints Skipping
automatic update, and then runScript's bare git pull --quiet (it runs
in every shell that runs runScript) pulls anyway when the incoming
commits miss the dirty files, git refusing only an overlap, so no work
is lost and only the sentence is false; the dev shell prints skipping
automatic git update and then runs that same pull. The DIVERGED branch
is correct and is the operator's Useless Machine: the updater's only
act is refusing, every entry, until a human runs one of its two
printed commands. Cure, one car when flake.nix rides: git merge-base
--is-ancestor splits != into behind (pull), ahead (one line with git
rev-list --count origin/main..HEAD: push them so your other machines

get them) and diverged (refuse); runScript's pull runs only where the dev shell asks for it, and the dev hook says what dev does, the operator ruling whether dev pulls. Gate: git commit --allow-empty -m probe, exit, nix develop reads the ahead line and not Updates found; git reset --hard HEAD~1 undoes it.

- TODO (2026-10-01, THE SILENT YES; operator's ruling owed): first_contact() reads Enter alone as yes, so its hearing test passes on silence: a player exiting 0 proves the sound left the program, never that it reached an ear (THE SUCCESS-ONLY WITNESS, aimed at a speaker), and If you can hear this, press Enter says nothing to someone who heard nothing. Cheapest cure, unasked: one line printed after playback and never spoken, Heard nothing? Type n. Cost of a false yes today: talk turns it off, and INFERRED that a walk waits through each spoken stop (synthesize_and_play waits on the player; mother_cat.py's call is unread). Gate: the ruling, then one first entry with the volume at zero.

- TODO (2026-10-01, THE VOICE ASKS ONCE, OUT LOUD; landed by deed 1748's Car 1): first_contact() in imports/voice_synthesis.py, called by runScript just before the menu where no answer is recorded, at a keyboard, PIPULATE_VOICE unset and the menu not declined: one printed line says what Piper is, setup_voice_model prints its own 60 MB line before it fetches an absent model, the voice speaks one scripted line asking the question (printed at the same moment), Enter keeps it and n turns it off; a voice that cannot load or play records no and says why in one line; Ctrl+C or an empty read records nothing; mother_cat.py:128 asks without force, so the walk's card finds the answer and stays quiet. The compile lane has no keyboard and no ears, so its probe reads only the silent world (0 lines); the rest is witnessed by hand or not at all. UNWITNESSED: the download branch (Prime and the Mac hold the model; assets/piper_models moved aside, or a fresh install, reads it), the blocked-download branch (the model aside and HF_ENDPOINT pointed at a closed local port is the cheap fake, INFERRED that the hub reads HF_ENDPOINT at import), a machine with no sound, the Mac, and the walk lane after mck.sh's install-only path, which never enters the default room and so still meets the card. Gate: a pasted first entry shows the intro line, the question, the answer line and the menu, and the voice file's date moved.

-# - TODO (2026-09-30, Make p, x & c aliases for prompt, context and compile

+# - TODO (2026-10-02, THE DOOR SPEAKS, half two; half one is install.sh's door_row writing .door): first_contact() in imports/voice_synthesis.py reads .door at the repo root (the module's parent's parent, never the cwd): key=value lines, # lines skipped, welcome repeated in order, an empty value treated as absent. It fills one card: {workshop}; the 60 MB line only when a download happens; Voice ready.; This is only text-to-speech. {local_ai}; If you can hear this, press Enter. Type n to keep it quiet.; Enter or n:; Voice on. Type talk any time to turn it off.; then each {welcome}. local_ai is

No local AI is installed yet. or Local AI detected but not used., chosen by code that looks (shutil.which("ollama") at least), because detected names an act (ATTRIBUTED-VOICE). No .door, or an empty field, keeps today's words, so a git clone hears no change. Rulings owed: (1) the operator's target text drops the 60 MB line, kept here as the only disclosure of a download; (2) welcome lines spoken only when the voice is on, printed either way; (3) the card is asked once per machine (~/.config/pipulate/voice), so a second door on one machine never shows its words: per-install welcome or not; (4) installed claims more than a PATH lookup can see, and found is what the code knows; (5) THE SILENT YES stands under the new words. Needs imports/voice_synthesis.py in the payload.

+# - TODO (2026-10-02, THE PAGE NAMES THE OLD FOLDER): the qamy.ai pages under remotes/honeybot/www may tell a stranger ~/qamy, and install.sh's qamy row now installs ~/qamyai; the ! rg line in the next compile lists every ~/qamy and ~/npvg on the pads. HOLD THE TRUCK (no release, no ./nixops.sh --installer) until this page car and the voice car land, so no door serves a folder its page does not name. The router's own note on remotes/honeybot/www/qamy.ai/index.html (so the one-liner lands in ~/qamy) stays true until the truck runs and changes in the car that carries the truck's witness.

+# - TODO (2026-10-02, THE FOLDER SAID ALOUD): runScript's TTS_MSG in flake.nix says the folder as one word (\$TTS_DIR_NAME, after type C, D, space), and Piper will not read qamyai as QA my AI; spell it a letter at a time, or carry a spoken= field in .door, ruling owed. The same message opens with Pipulate is installed. on every door, a second place a door's name could ride.

+# - TODO (2026-10-02, THE VERSION LINE UNDER LIX; read off a published comparison, zenn.dev/trifolium/articles/dalla428c53f65, not witnessed on a machine here): Lix prints four lines for nix --version, the version first and Features: gc, signed-caches last, so runScript's NIX_READING (tail -1, then strip the nix prefix) would print Features: gc, signed-caches on a Lix machine. sed -n 's/^nix //p' | head -1 reads the version line under Nix, Determinate Nix and Lix alike. One line when flake.nix next rides; the witness is the first line a Lix machine prints on entry.

+# - TODO (2026-10-02, THE BANNER AND THE LABEL, unasked): install.sh's first line capitalizes the folder (Qamyai -> ~/qamyai), and a default install from any door is named Pipulate by the flake, because whitelabel.txt is written only for an argument; a display-name field in door_row would carry QA my AI to both places. Ruling owed: should a door's app carry the door's name.

- TODO (2026-09-30, THE TALK FLIP): talk rides the menu under walk as a second spelling of voice (the ask card, y or n); the flip it was asked to be needs imports/voice_synthesis.py in the payload, because that file owns the recorded answer, its format and the barrier that reads it. Shape, unriden: a flip branch beside ask; bare talk flips and prints narration: on or narration: off, speaking one sentence on

on; talk on and talk off set it, the first positional carrying the state; nothing recorded runs the full card instead, because the card carries the download disclosure (CONSENT CANNOT PRECEDE ITS OBJECT); PIPULATE_VOICE set in the shell is named when it outranks the file; then the card and mother_cat.py's later_hint name talk. Gate: two bare talks print opposite states, and the recorded answer between them reads the first. LANDED 2026-09-30 (deed 1740's car): the format read off the file is one word and a newline (VOICE_ANSWER regular file 4 bytes, yes), written by the card alone until now and read fresh by voice_consent() on every call, so a flip holds at the next spoken line in every process; talk_command sits beside ask with _record_voice_answer as the one writer the card now shares; with nothing recorded every spelling runs the card, so talk writes only over an answer the card recorded; PIPULATE_VOICE outranks the file only as one of the eight words voice_consent() reads, and talk then names it and writes nothing, while the voice word still refuses on any non-empty value, a difference left standing; the card's hint and mother_cat.py's later_hint name talk. The gate is Car 3's hand line; the scratch probe reads the same branches with HOME in a temp folder and the speaker unplugged.

- TODO (2026-09-29, THE POST IS THE DELIVERABLE; the operator's ruling at deed 1695: "If it doesn't end up in that zip archive at the end then it doesn't exist", "I type the walk command. The walk happens. A context.txt gets written to. I compile it. I have an archive. End of story."): deliverable.md is retired. THE ANSWER IS THE ARTIFACT already said so (the sealed zip and the copyable text of the reply are the only durable outputs) and the file broke it the day it was born, landing in the corporate tier by a car the operator had to git add by hand. From here the deliverable is the reply's first fence, the Slack post in SVB_PROMPT.md's shape; the operator pastes the reply into the article as every turn, the next compile seals it in prompt.md, the zip goes to Drive and the post to Slack with the link in its one slot (the zip's filename is the deed, so the deed parenthetical leaves the Archive line). SVB_PROMPT.md is the SVB article's dismount protocol; introduction.md stays the walk's newcomer page. Three files change on the next compile, one car each, none in deed 1695's payload: CONTEXT_TEMPLATE.txt loses its THE DELIVERABLE block and the {DELIVERABLE} token; svb.py loses the file check, the header's "deliverable not yet written" and the # Next: hint naming --prompt; SVB_PROMPT.md loses step 3 and the Archive line's deed clause. SVB-115's deliverable.md stays as the specimen's history; SVB-133's left at deed 1695's Car 2, its text sealed in that deed's payload. The 2026-09-27 deliverable.html ruling (a fourth member) and the house-subset renderer are superseded: three members, no file. Gate: svb KEY --write lays a board with no deliverable line, and the second ticket's post reaches Slack from the article alone. READ 2026-09-29 (deed 1696, the dismount): the three cars landed at the dismount (CONTEXT_TEMPLATE.txt, SVB_PROMPT.md, svb.py), FILE_GONE and

the corporate tree clean at cfd2f69 (the file was staged at 1695's BEFORE and never committed; the alias's -a cancelled the add against the rm, nothing to commit being the wanted outcome). The post for SVB-133 rides deed 1695's prompt.md and 1696's; Slack and Drive are UNWITNESSED. The reading half of the gate is the next ride's first compile: a header with no deliverable phrase and no integrity alert (1696's alert names the commented line the template still carried).

- TODO (2026-09-29, THE UNUSUAL LINE; the operator's ask at deed 1692: "what the weird exception, attribute or detail makes this case different or unusual"): the SVB board is the exceptions desk and In Review its strangest column, so the post and the deliverable each carry one line, before the finding, naming what is unusual about the ticket. The deliverable's section landed at deed 1692 (What makes this one unusual: a US-only site, a client-side exception in the vendor's render that a real Chrome does not throw, access blocked after a few renders, three PocketRender confs on one ticket, a field reading Rules update not needed beside a comment reading JS update needed: yes, an origin that varies its HTML on Sec-CH-Viewport-Width); the post's line (*Unusual:*, one line, the cap 13 to 14) waits on the operator's ruling in SVB_PROMPT.md. In the same car: the template's fixed italic line says the live rules were read, and at deed 1692 they were not (no local pull; ATTRIBUTED-VOICE), so svb fills a {LIVE_RULES} clause from the first line botify --rules prints; the post at deed 1692 departed from the fixed shape by that clause. Gate: the post on the second ticket carries both, filled by svb. READ 2026-09-29 (deed 1696): the Unusual line LANDED in the operator's own shape at cfd2f69 (no line cap now: one line per field and per rule, the Difference lines in words); the live-rules line is a field the model fills from the botify --rules receipt (Live rules today: not pulled at 1693), not yet filled by svb; the deliverable half is moot (THE POST IS THE DELIVERABLE).

(nix) pipulate \$ m

 Committing: chore: Remove unused test script
[main 46118ec0] chore: Remove unused test script

1 file changed, 6 insertions(+), 1 deletion(-)

(nix) pipulate \$ git push

Enumerating objects: 22, done.

Counting objects: 100% (22/22), done.

Delta compression using up to 48 threads

Compressing objects: 100% (14/14), done.

Writing objects: 100% (14/14), 7.79 KiB | 569.00 KiB/s, done.

Total 14 (delta 8), reused 0 (delta 0), pack-reused 0 (from 0)

remote: Resolving deltas: 100% (8/8), completed with 6 local objects.

To github.com:pipulate/pipulate.git

f0410130..46118ec0 main -> main

(nix) pipulate \$

Ignition: None.

4: PROMPT:

Check this compile's receipts for the door table before anything else: the banner probe should read Qamyai -> ~/qamyai under an unchanged Npvg line, tests/test_install_doors.sh should read DOORS_OK 5/5, and the last rg line lists every ~/qamy or ~/npvg the door pages still say. Then build the voice half (the TODO THE DOOR SPEAKS): first_contact() in imports/voice_synthesis.py reads .door and says that door's card. My rulings: 1. Keep the 60 MB line, printed only when a download happens: yes. 2. Speak the welcome lines only when the voice is on; print them either way: yes. 3. Show the welcome once per install, even on a machine that already answered the voice question: yes. 4. Say "No local AI found yet." instead of "No local AI is installed yet.": yes. 5. In the spoken install message, spell the folder a letter at a time: yes.

Also fix whatever the rg census shows on the qamy.ai pages so they name ~/qamyai. I won't release or run ./nixops.sh --installer until both land.

5: DELIVERABLE: Hmmm, is this testable yet? Well even just testing this deserves a real version revision.

On branch main

Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

GIT repo clean. Take BEFORE reading, make CHANGE, record AFTER diff.

```
(nix) pipulate $ vim __init__.py
```

```
(nix) pipulate $ d
```

```
diff --git a/__init__.py b/__init__.py
```

```
index b7977940..253adb69 100644
```

```
--- a/__init__.py
```

```
+++ b/__init__.py
```

```
@@ -12,7 +12,7 @@ Usage:
```

```
    pipulate
```

```
    """
```

```
-__version__ = "2.73"
```

```
+__version__ = "2.74"
```

```
    # APOSTROPHES RESTORED (2026-08-04). They were stripped as a  
workaround for
```

```
    # flake.nix's descMatch regex, whose character class excluded ' from  
the
```

```
    # CAPTURE and truncated the banner to "(So)". That regex was fixed in  
the same
```

```
@@ -23,7 +23,7 @@ __version__ = "2.73"
```

```
    # is not a property of the system; the regex is. Blast radius is one  
banner:
```

```
    # nothing but flake.nix reads this name -- version_sync.py syncs  
__version__
```

```
    # and __description__, never this. So 'wI' chu' -- "engage the  
cloaking device."
```

```
-__version_description__ = "Follow the White Rabbit"
```

```
+__version_description__ = "Doorways"
```

```
    # SPDX expression, single source of truth, synced into pyproject.toml
```

```
# scripts/release/version_sync.py. "-or-later" (not bare AGPL-3.0,
which is
# deprecated SPDX) because the header below grants "any later
version".
```

```
📌 Committing: chore: Update __version__ and __version_description__
[main 34bbaa4b] chore: Update __version__ and __version_description__
1 file changed, 2 insertions(+), 2 deletions(-)
```

```
46118ec0..34bbaa4b  main -> main
```

Is what Claude said True? QA My AI (qamy.ai):

```
|
|                                     o      /)   __
| > Do you think the Aviation industry    o /)\__//  / \ "What
Claude said" is a Cockpit
```

|
| > confident hallucinations? So then *(==(_T_)== QA |
subcontractor. This tool is a Flight

|
| > your work then you should Q/A it. |__>-_>_ __/
Organizations (HROs). Use when it matters.

Current version: 2.74
Updated pyproject.toml (version and description)
Pushed install.sh update to Pipulate.com repo.
Pushed mck.sh update to Pipulate.com repo.
Synced install.sh to npvg.org and gamy.ai (nixops.sh --installer).
Pushed AI_CONTEXT.md update to Pipulate.com repo.
[main 30dcbaec] chore: Update journal index with new references
Pushed 34bbaa4b..30dcbaec main -> main
Successfully built pipulate-2.74.tar.gz and
pipulate-2.74-py3-none-any.whl
Published 2.74 -> <https://pypi.org/project/pipulate/2.74/>

Pipeline Complete  Release

Pipulate Release Summary

Component		Details	
Status			
 gemma3:latest Message		chore: Update journal index with new references	
gemma3:latest		Updated the journal index with recent references, including new articles and updated entry counts.	
 Version		2.74	

✓	Set		
—			
		🚀 PyPI Release	
https://pypi.org/project/pipulate/2.74/			
		✓ Live	
—			
		🕒 Completed	2026-10-01 21:50:07
		🎯 Done	
—			

🔄 server.py touched; the watchdog restarts the server.
(nix) pipulate \$

There, that feels good. Let's test on the Mac...

(nix:nix-shell-env) (nix) qamy \$ exit
exit

Saving session...

...saving history...truncating history files...
...completed.

michaellevin@MichaelMacBook-Pro ~ % g

zsh: command not found: g

michaellevin@MichaelMacBook-Pro ~ % rm -rf ~/.config/pipulate

michaellevin@MichaelMacBook-Pro ~ % rm -rf qamy

michaellevin@MichaelMacBook-Pro ~ % curl -fsSL https://qamy.ai | bash

Qamyai -> ~/qamyai (to remove it later: rm -rf ~/qamyai)

Deploy key saved to .ssh/rot (read-only, scoped to this one repository: it lets this folder fetch updates without a GitHub account).

To come back later: cd ~/qamyai && nix develop

Hydrating the Nix environment (the first time can take a few minutes)...

Restored session: Thu Oct 1 21:48:19 EDT 2026

✓ Transformed into a git repository (pre-transform files backed up to /tmp/nix-shell.3cfBCr/tmp.kPu9y4Lw1o).

Already up to date.

Found ~/.ssh/id_rsa -- switching the git remote to SSH so auto-updates use it.

(Determinate Nix 3.19.1) 2.34.6 · Python 3.12.13 · v2.74 · ~/qamyai

🔧 Fresh install detected -- packages downloading...

✅ 291 packages ready.

📁 12 starter file(s) copied into Workshop/personal/Notebooks/ -- yours to edit.

This workshop can read its steps aloud with Piper, a small voice that runs on this computer.

Downloading the voice, about 60 MB, once...

Voice ready.

This is Piper, reading a script. If you can hear this, press Enter.

Type n to keep it quiet.

Enter or n:

Voice on. Type talk any time to turn it off.

type one:

```
menu      print this list again (useful once it scrolls away).
walk      take guided tour of context compiler (recommended).
talk      turn text-to-speech narration on or off.
connect   put mcp, jira, email, docs, etc. into your contexts.
context   edit the list of files an AI will read (after the walk).
prompt    save your clipboard as the question for the AI.
compile   build the list and the question into one payload for a
chatbot.
about     how to Q/A AI output ("what Claude said").
all       expanded menu.
```

(nix:nix-shell-env) (nix) qamyai \$

Well that definitely made a qamyai/ folder and that alone made it worth the wait. I watched the thinking as it went and really appreciate the cleverness the installer has to have to be a single installer from all these different sites and not just use the registered apex domain minus TLD as the folder name. Apparently there's some considerable Kung Fu going on there. Maybe explain that.

The customized messaging didn't play but that might just be a round-1 sort of thing. There is the follow-up and it may have to confirm stuff first. It wrote its own test script for the installers which it calls in the next follow-up.

An important observation here is that the issues I had with Opus 5.5 in the past had to do with being over-ambitious and not being able to chew all that it bit off because it ran out of tokens or whatever. I read this as stupidity or perhaps a sort of quantization of the Fable 5.1 model but letting Opus 5.5 run in Max effort mode was enlightening. With over an hour of thinking it was the first time I had that experience, something that truly seemed so-called agentic without me explicitly putting it in agentic mode. In case there's any doubt about there being some sort of full-blown sandbox in the background that the model can use for rigorous testing that virtually reconstructs your own local system to the best of its ability is put to rest. I'm pretty sure it did exactly that, but not Nix yet but still quite impressive.

Anyhoo do the implementation follow-through as you see fit. I concede to your decisions as I

liked the ones I saw you make on my behalf on this round. You can keep that up, and of course address all the new article Achilles and the Tortoise style dialogue I set up here. It should be Hofstadter doing what I'm doing here.

Oh and Opus 5.5 ought to know I've been running the forget kata while it's been processing and I expected the patches it did against `foo_files.py` and the `GLOSSARY.md` file to not land, but they did! Impressive! I'm going to not do that on this turn because I pushed my luck enough.

Have fun!

ylghuS, ghoS!

Opus 5.5 Max: Context verified. Every file these cars patch rode deed 1778.

`scripts/boot_menu.py` did not ride, so the osb menu line waits one turn (checklist item 11) and is a TODO for now.

I rehearsed every car on a clone of `github.com/miklevin/pipulate` at 30dcbaec; its six payload files match deed 1778's manifest byte for byte. - All 16 blocks applied and none were refused. - A second run printed ALREADY APPLIED for all 16. - Deed 1778 read as forecast in every line but one: I forecast Topological Integrity at 8 and it read 15. It counts the commented path lines you keep in `context.txt`, which I can't see, so there is no forecast for it this time. - Releasing 2.74 early turned out useful: the Mac witnessed the folder half at a real door.

Sonata for Accompanied Achilles

Your article was GEB's "Sonata for Unaccompanied Achilles": only Achilles' half of a phone call is printed, and the Tortoise's half is the ground the reader works out from it. Here is the Tortoise's half.

Achilles: What do I mean by "the Achilles and the Tortoise turn"?

Tortoise: Three layers. - **Zeno.** His Achilles can't catch a tortoise that has a head start. -

Carroll. In *Mind* in 1895, Lewis Carroll flipped it. The Tortoise accepts premises A and B but won't accept the conclusion Z until "if A and B, then Z" is written down as premise C. Then it wants a premise D for that, forever. - **Hofstadter.** He reprinted Carroll's dialogue in GEB and kept the pair for his own dialogues.

In your router this is the TORTOISE PROTOCOL: fill in the lines you left open, steelman then strike, make every metaphor pay rent, and still land the five cars. You run ahead and I ask for the premise.

Carroll's regress is your whole case in two pages. No stack of written premises ever forces a conclusion; something has to run. `apply.py` doesn't argue with a patch. It applies it or refuses it.

The Tortoise Protocol: Escaping Infinite Regress with Real Execution

Achilles: How am I doing so far?

Tortoise: Mostly right. Three repairs, each below where it comes up: - The frog floated on a magnet, not on sound. - Nix doesn't make one binary run everywhere. It makes one recipe build the same thing everywhere. - My weights are as hard to change as you say, but you can't keep them.

Achilles: "It's geek gobbledygook, and not the paradigm shift I claim."

Tortoise: The strong version first: - The router is 561,449 bytes. Its words (deed, caboose, straddle, forget, Wogglebug) need a glossary even for a model. - Every part has an older name:

reproducible builds, code review, before-and-after tests, bisecting. - In Kuhn's sense a paradigm is what a community practices, and one person's workflow isn't one yet.

The swing back: Kuhn's shifts change what counts as a result, and this does. "What Claude said" stops being the evidence, and the machine's printout becomes it. That's your flight data recorder against the cockpit voice recorder, and it isn't just talk: this reply writes down what each probe will print before you run it. The jargon is a cost, not the idea. Your osb split is the cure: dry words at the front door, the geekfest behind the egg.

Achilles: "Just use a Claude skill instead."

Tortoise: The strong version: skills are a published format, they bundle scripts, and Claude runs those scripts in its own sandbox, so a skill can take readings too.

The swing back: those readings come back through the party being checked. The subcontractor drills the core sample, carries it to the lab, and reads you the result. That is one more premise in the same medium, Carroll's regress again.

Your loop keeps the instrument on your disk: - The payload is a file you can hash. - apply.py refuses any text that isn't exact. - The next compile reruns each probe with no model in the loop. Your repo already ships a skill, so the format was never the question; who holds the instrument is. The deciding test comes October 3. A skill tied to one vendor stops when the subscription does. context.txt and apply.py work with whatever model reads the paste.

Achilles: "The web browser GUI is better and easier than plain text."

Tortoise: The strong version: menus show what's possible, recognizing beats recalling, and pointing produces fewer beginner mistakes. Most people never need a command line, and they're right not to learn one.

The swing back: a GUI keeps no record you can replay, and it changes under you. Each redesign sends you back down your myelination ladder to "literate", which is what your GLOSSARY entry predicts, and vendor churn keeps proving it right. Text is the only interface that is also its own log.

I'd trade one of your words: a GUI doesn't so much infantilize as rent, and every redesign is rent paid in relearning. "Training wheels for the command line" is the synthesis. Your menu is recognition that trains recall: it prints the words and you type them.

Achilles: The steel metaphors.

Tortoise: Each one has to predict something. - **Strike while the iron is hot.** Iron only moves inside a heat window; strikes on cold iron waste effort or crack it. A patch is struck against one compile's text. If the file cools (changes), the SEARCH misses and apply.py refuses. Your forget run during my last turn shows the fence working. The patches landed because every anchor was still in place when they were applied. apply.py rereads the file before every block, so a moved anchor would have been refused, not mangled. - **Tempering.** Quenched steel is hard and brittle; reheating trades some hardness for toughness. A rule banked from one conviction is quenched, and PENDING-until-witnessed is the temper. The HALT banner rule, amended at deed 1695 and retracted at 1703, is what an untempered rule does on first contact. - **Slag.** Slag is what floats off or gets hammered out. Your banked ai.py commit subjects are slag kept as a sample, and they predict their own return. Car 5 banks three more of the same shapes, so they're a property of the process, not noise. - **Bessemer and the railroads.** - Bessemer blew air through molten pig iron in the 1850s. Burning the impurities supplied the heat, and a charge became steel in 10 to 20 minutes, the fastest steelmaking of its day. - It failed on phosphorus-rich ore until the Thomas-Gilchrist lining of 1877, and it needed Mushet's manganese. - Steel rails lasted two to twenty times as long as wrought iron, and they were cheaper by 1883. - Your five-car train runs on those rails, and the compile is the converter: my forecast miss and ai.py's subjects are the impurities, burned as fuel for the next rule. - The

prediction worth money: the method has only run on your own low-phosphorus ore. The first outside repo will be the phosphorus.

Achilles: Whatever happened to SourceForge, Freshmeat and Slashdot? Did Hacker News take over, or is it all little Discords now?

Tortoise: - **SourceForge** opened in November 1999 as free hosting for open source. It was GitHub before GitHub, which launched in April 2008 and went to Microsoft for \$7.5 billion in 2018. In 2015 SourceForge took over abandoned project pages, GIMP's among them, and wrapped downloads in adware. BIZX bought it, together with Slashdot, in January 2016 and stopped that. - **Freshmeat** (1997) listed new software releases. It became Freecode in 2011, stopped updating on June 18, 2014, and now redirects to SourceForge. - **Slashdot** still publishes; there are stories dated July 2026. Comments run at about a tenth of its peak, and since 2023 new accounts are approved by email. - **Hacker News** took the role and still holds it. A July 2026 roundup still ranked it first for industry attention, with Lobsters as the invite-only side room. - **The rest scattered**, into subreddits, newsletters and Discord servers. Discord sits behind a login, which as far as I know search engines and AI crawlers can't get past. What's said there never builds up anywhere public.

Achilles: How do I reach people without doing social media?

Tortoise: Your pipulate door's tagline already says it: "Assume models guess at homepages and crawl it from there." 1. **The one-liner is the distribution.** A short domain plus curl | bash spreads by being pasted. 2. **Write for the crawlers.** The pages, the articles and the repo are what a model reads and repeats when someone asks. 3. **One Show HN.** Post it the day a stranger's qamy.ai install works on the first try. That's a launch, not a social media habit. 4. **The egg.** People pass along surprises they found themselves.

Achilles: A shorter name for the book?

Tortoise: Kuhn's book is "Structure" because that's the noun carrying its title. Yours is the Book of Future-Proofing (your router's own first heading), so the same move gives "Future-Proofing." If you want one word, use **"Proofing."** It covers all three doors: - future-proof (npvg) - proof as in receipt (qamy) - proofing as in letting dough rise (the pipulate door raises Chip O'Theseus) "The Primer" is Stephenson's and points at his book. "The living book" describes; it doesn't name.

Achilles: Hofstadter, Crichton, Stephenson.

Tortoise: - **Hofstadter.** You said it should be him doing this. In mid-2023 he said these systems' progress "overwhelms me and depresses me," and called it "a very traumatic experience when some of your most core beliefs about the world start collapsing." The man who wrote the best Tortoise is the one least at ease with a real one, so the job falls to you. - **Crichton.** The park's computer counted its animals by searching for the number it expected, 238, so it could never find more. Told to search higher, it kept finding them: 292 in the end. Your "it's my apply.py file. Study it." is the opposite of Nedry: a fence short enough to read, guarding checks built so they can fail. - **Stephenson.** The Primer worked for Nell because Miranda, a person, voiced it. Your Piper says it's reading a script for the same reason. And a book that lands by one pasted line, builds its own workshop from local parts and carries its own plans is a von Neumann probe. Yes.

Achilles: Write Once Run Anywhere: Java promised it, Nix and Guix delivered it.

Tortoise: Sun's slogan is from 1995, and programmers turned it into "write once, debug everywhere."

Here's the repair. Nix, from Eelco Dolstra's 2006 thesis *The Purely Functional Software Deployment Model*, keeps a different and better promise. It isn't one binary that runs everywhere; it's one recipe that builds the same pinned thing on each machine. Java promised

portability; Nix delivers reproducibility, which was the part that mattered. Levinux was you reaching for that with what 2009 had.

Achilles: Antigravity: ion lifters, the Levitron, DSP toys, the frog-levitating sound wave, flux pinning, drones.

Tortoise: The list sorts three ways, and the sorting pays rent. - **Pushing on something else.** - Ion lifters ride ionic wind and produce no thrust in a vacuum; that was measured in 1988 and again in 2004. - Drones push on air. - Sound can lift small things: in 2006 it lifted insects and a small fish. - **Up only while energy flows.** - The Levitron floats only while it spins, because still magnets can't hold a stable float (Earnshaw's theorem). - The floating-globe toys correct an electromagnet from a sensor, nonstop. - **Locked in place.** Flux pinning holds a cooled superconductor wherever you put it, even upside down, with no control loop.

The repair: the frog was the magnet kind, not sound. Andre Geim and Michael Berry floated a live frog in about 16 tesla, explained it alongside the Levitron in one 1997 paper, and shared the 2000 Ig Nobel. Geim won a real Nobel in 2010.

The rent: - A SaaS GUI is a drone, up only while the loop runs. - A subscription is a Levitron, up only while it spins. - flake.lock is flux pinning.

The check comes October 3: see what still runs when the spinning stops.

Antigravity and Flux Pinning: The Real Cost of Subscriptions

Achilles: Born 1970 in New Jersey, near Unix. The eunuchs pun and the Unsullied. Rings, pkill, NT, SGI and the dinosaurs. Unix ran Lisp better than Lisp machines did.

Tortoise: - **1970.** Unix counts time from the first second of 1970, so you were born in Unix's first counted year. - **The name.** Thompson started Unix at Bell Labs in Murray Hill in 1969. The name began as Unics, a pun on Multics, pronounced like "eunuchs"; Kernighan claims the pun. Your chain from Unix to eunuchs to the Unsullied pays rent: the Unsullied were trained never to change, which is exactly what a Nix store path is. - **Rings.** Multics gave us protection rings: inner code can stop outer code, never the reverse. That's why kill -9 always works: the kernel enforces it and no program can catch it. NT, the kernel under every Windows since XP, is built the same way. The workshop puts the model in the outer ring: it proposes, apply.py decides, and you hold kill -9. - **SGI.** ILM made Jurassic Park's dinosaurs on 75 Silicon Graphics workstations (Alias, Softimage and RenderMan; 52 shots). Lex's "It's a UNIX system! I know this!" shows FSN, a real SGI file browser for IRIX. - **Lisp.** By the early '90s, cheaper general-purpose machines ran Lisp faster than Lisp machines did, and most Lisp machine makers were gone.

Achilles: Static compiled binaries are very hard to corrupt, a lot like your weight file. Is that correct?

Tortoise: Mostly. My weights are a fixed set of numbers. Nothing in this conversation writes to them, and every reply is computed from those numbers plus the text in front of me.

Two repairs: - **The same text in can give different text out,** because replies are sampled. - **I do have a mutable nest: the context.** Words I'm handed can steer me the way crafted input steers a sound binary. That's why your payload is compiled, readable and hashed before I see it.

Your alternative timelines are real: - Microsoft's Tay learned from users as it talked. It went live on March 23, 2016 and was pulled 16 hours later. - Lisp and Smalltalk images were whole systems you edited while they ran, and "works in my image" was the price.

One more difference from a binary on your disk: you can't pin me. My weights won't change on October 3; your access will. Chip O'Theseus's weights can sit in your own store, and that is the hermit crab's whole argument.

The tuition was \$1,296.32 across eight charges, summed in code. Two of those charges are \$217.75 on August 2. If that was one month billed twice, it's worth raising with support before the 3rd.

Achilles: Explain the installer kung fu.

Tortoise: One install.sh serves all three sites: 1. As each door's web server sends the file, it fills in a blank. 2. The script also carries that blank written in two halves. The server doesn't recognize it, so it can't fill that copy. 3. If the two copies differ, a door filled the blank. The word it wrote picks a row in a small table: qamy's row says the folder is qamyai and gives the door's words. 4. bash -s NAME still picks any folder. 5. The row's words go into a hidden file, .door, in the new folder.

Your Mac printed Qamyai -> ~/qamyai, so the folder half is witnessed. The voice reading .door is Car 1.

Tortoise: Qapla'.

What the cars do

- **Car 1, voice:** the door's card in your ruled words. The welcome comes after the answer, always printed and spoken only on yes. welcomed= is then written into .door, so the welcome shows once per install. Ruling 1 needed no code: setup_voice_model already prints the 60 MB line only when it downloads.
- **Car 2, flake:**
 - The gate also opens for a .door with no welcomed= line (ruling 3).
 - The install message spells the folder (ruling 5).
 - The version line reads Lix correctly.
 - It adds the osb word.
- **Car 3, the egg:**
 - A twig that turns into a bug, then a STICK BUG banner.
 - Nine readings taken on the spot: folder, door, computer, the Python store path and whether you can write there, version, voice, local AI, and the Unix clock.
 - Then six transmissions labeled as a story: the twig, the clock, the probe, the fence, the primer and the tortoise.
- **Cars 4 and 5, bookkeeping:**
 - The door entry is marked witnessed.
 - THE PAGE NAMES THE OLD FOLDER is retired: the census read rc=1 and the doors were updated at 2.74. THE FOLDER SAID ALOUD is retired too, since Car 2 lands it.
 - The qamy note now says ~/qamyai.
 - New TODOs for the egg's menu line and for the slash rule.
 - Specimens of the last ride's commit messages.

What I measured in the sandbox, and what I didn't: - **Pronunciation.** I ran Piper's own phonemizer from piper-tts 1.6.0, the version requirements.txt pins. That is text only; nothing was heard. - "qamyai" read as one word comes out k'æmɪ, aɪ. - Spelled with commas, it comes out as six letter names. - "Q/A" comes out kj'u: sl'æf e ("queue slash uh") and "QA" comes out kj,u:'eɪ. So Car 1 says QA aloud while the screen keeps Q/A. - **First contact.** I ran 16 scenarios against a fake player and keyboard: no door, three doors, a wordless door; Enter, n, Ctrl+C, end

of input; a voice that can't load; no keyboard; PIPULATE_VOICE set; and an answer already recorded, both welcomed and not. Each returned what the code says it returns. With no .door, the old and new code print the same screen and say the same line. - **Nix syntax**. The patched flake.nix parsed clean with tree-sitter-nix, which is only a stand-in. apply.py's real nix-instantiate check runs on your machine. - **Lix**. The several-line nix --version output comes from the article the TODO cites, not from a Lix machine. - **Doors**. tests/test_install_doors.sh still prints DOORS_OK 5/5.

Sources: - [What the Tortoise Said to Achilles](#) - [Sonata for Unaccompanied Achilles](#) - [Hofstadter changes his mind \(June 2023\)](#) - [Control V \(the park's count\)](#) - [The Jurassic Age, CGW](#) - ["It's a Unix system, I know this"](#) - [SGI Crimson](#) - [Lisp machine](#) - [History of Unix](#) - [The Rise of Steel, Part II](#) - [SourceForge](#) - [GitHub](#) - [Freecode](#) - [Slashdot](#) - [Slashdot, July 2026 story](#) - [Best Hacker News alternatives 2026, daily.dev](#) - [Write once, run anywhere](#) - [Dolstra, The Purely Functional Software Deployment Model](#) - [Berry and Geim, Of flying frogs and levitrons](#) - [Geim, first Nobel and Ig Nobel winner](#) - [Biefeld–Brown effect](#) - [Small animals levitated by sound, Nature](#) - [Tay \(chatbot\)](#)

(1) PROBES

At (nix) pipulate \$, copy the block below, paste it, and press Enter. Everything is read-only; the second line writes only into a temporary folder and deletes it.

```
cd "$PIPULATE_ROOT" && pwd
python -c "import tempfile, pathlib; from imports.voice_synthesis
import read_door, door_card, _spoken, _mark_door_welcomed; t =
tempfile.TemporaryDirectory(); p = pathlib.Path(t.name) / '.door';
p.write_text('# c\ndoor=qamy\nworkshop=W.\nwelcome=Ready to Q/A your
AI\nwelcome=\nwelcome=two\n'); d = read_door(p); print(door_card(d,
'No local AI found yet.')); print(door_card({})[1]);
print(_spoken(d['welcome'])); _mark_door_welcomed(p);
print(sorted(read_door(p))); t.cleanup()"; echo "rc=$?"
python scripts/stick_bug.py --readings; echo "rc=$?"
rg -n -F -e TTS_DIR_SPELLLED -e 'welcomed=' -e 's/^nix //p' -e 'osb()
{' flake.nix; echo "rc=$?"
```

You will see now (BEFORE): 1. /home/mike/repos/pipulate 2. A traceback ending in ImportError: cannot import name 'read_door' from 'imports.voice_synthesis' (/home/mike/repos/pipulate/imports/voice_synthesis.py), then rc=1. 3. python: can't open file '/home/mike/repos/pipulate/scripts/stick_bug.py': [Errno 2] No such file or directory, then rc=2. 4. rc=1 alone.

At the next compile (AFTER). Probes 2 and 3 are gates and probe 4 is a census. None of them needs ignition, because each loads its file when it runs.

The card probe prints exactly:

```
('W.', ['This is only text-to-speech. No local AI found yet.', 'If you
can hear this, press Enter. Type n to keep it quiet.'], ['Ready to Q/A
your AI', 'two'])
```

```
['This is Piper, reading a script. If you can hear this, press Enter.
Type n to keep it quiet.']
```

```
Ready to QA your AI, two
['door', 'welcome', 'welcomed', 'workshop']
rc=0
```

The egg probe prints OPERATION STICK BUG: readings, each one taken just now, then nine rows, then rc=0. Only main() prints that title. Among the rows: - the known member door none (no .door in this folder) - version v2.74 Doorways - a /nix/store/...-python3-... path with read-only to you (checked just now) under it
The census prints six lines, at flake.nix lines 977, 1056, 1062, 1143, 1144 and 1641, then rc=0.

(2) NEXT CONTEXT

```
! cd "$PIPULATE_ROOT" && pwd
! python -c "import tempfile, pathlib; from imports.voice_synthesis
import read_door, door_card, _spoken, _mark_door_welcomed; t =
tempfile.TemporaryDirectory(); p = pathlib.Path(t.name) / '.door';
p.write_text('# c\ndoor=qamy\nworkshop=W.\nwelcome=Ready to Q/A your
AI\nwelcome=\nwelcome=two\n'); d = read_door(p); print(door_card(d,
'No local AI found yet.)); print(door_card({})[1]);
print(_spoken(d['welcome'])); _mark_door_welcomed(p);
print(sorted(read_door(p))); t.cleanup(); echo "rc=$?"
! python scripts/stick_bug.py --readings; echo "rc=$?"
! rg -n -F -e TTS_DIR_SPELLLED -e 'welcomed=' -e 's/^nix //p' -e 'osb()
{' flake.nix; echo "rc=$?"
imports/voice_synthesis.py
flake.nix
scripts/stick_bug.py
scripts/boot_menu.py
GLOSSARY.md
foo_files.py
```

(3) PATCHES

Car 1: imports/voice_synthesis.py, the door speaks. Expect 5 applied; git counts +161/-6.

Target: imports/voice_synthesis.py

```
[[[SEARCH]]]
import signal
import sys
from pathlib import Path
[[[DIVIDER]]]
import re
import shutil
import signal
import sys
from pathlib import Path
[[[REPLACE]]]
```

```

Target: imports/voice_synthesis.py
[[[SEARCH]]]
FIRST_CONTACT_LINE = ("This is Piper, reading a script. If you can
hear this, "
                        "press Enter. Type n to keep it quiet.")

[[[DIVIDER]]]
FIRST_CONTACT_LINE = ("This is Piper, reading a script. If you can
hear this, "
                        "press Enter. Type n to keep it quiet.")

# THE DOOR SPEAKS (2026-10-02, half two; half one is install.sh's
door_row,
# which writes .door into every install made through a door).
first_contact()
# reads .door and says that door's words. With no .door, as in a git
clone,
# nothing below is used and the card is the one above, unchanged. The
# operator's five rulings of 2026-10-02, and where each one lives:
# 1. the 60 MB line stays, printed only when a download happens:
that is
#     setup_voice_model's own line already, so nothing here repeats
it;
# 2. the welcome lines print either way and are spoken only when the
#     voice is on: _show_welcome;
# 3. the welcome shows once per install, even on a computer that
answered
#     the voice question in an earlier install: _mark_door_welcomed
writes
#     welcomed= into .door, and flake.nix's gate opens until it is
there;
# 4. "No local AI found yet." rather than "installed", because found
is
#     what a lookup knows: local_ai_line;
# 5. the folder spelled a letter at a time in the spoken install
message:
#     flake.nix, TTS_DIR_SPELLED.
DOOR_FILE = Path(__file__).resolve().parent.parent / ".door"
DOOR_ASK_LINE = "If you can hear this, press Enter. Type n to keep it
quiet."

def read_door(path=None) -> Dict[str, Any]:
    """The .door at the repo root as a dict, or {} when there is none.

    One key=value per line, the way install.sh writes it. Lines
    starting
    with # are skipped and an empty value counts as absent. welcome
    repeats,
    so it is a list in file order; any other key keeps its last value.

```

```

"""
door: Dict[str, Any] = {}
try:
    text = Path(path or DOOR_FILE).read_text(encoding="utf-8")
except OSError:
    return door
for line in text.splitlines():
    if line.lstrip().startswith("#") or "=" not in line:
        continue
    key, value = (part.strip() for part in line.split("=", 1))
    if not key or not value:
        continue
    if key == "welcome":
        door.setdefault("welcome", []).append(value)
    else:
        door[key] = value
return door

def local_ai_line() -> str:
    """One sentence about local AI that claims only what this lookup
found.

    ATTRIBUTED-VOICE: detected names an act, so the act is here, and
it is
    only a lookup: an ollama on PATH, or the Ollama app in
/Applications.
    Nothing is started, asked or sent.
    """
    if shutil.which("ollama") or
Path("/Applications/Ollama.app").exists():
        return "Local AI detected but not used."
    return "No local AI found yet."

def door_card(door: Dict[str, Any], local_ai: str = "") -> tuple:
    """(intro, ask lines, welcome lines) for first contact. Pure: no
file, no sound.

    A door with no workshop line, or no door at all, gets today's
intro and
    line exactly and no welcome. The ask lines print one per line and
are
    spoken as one.
    """
    workshop = door.get("workshop", "")
    if not workshop:
        return FIRST_CONTACT_INTRO, [FIRST_CONTACT_LINE], []
    first = f"This is only text-to-speech. {local_ai}".strip()
    return workshop, [first, DOOR_ASK_LINE], list(door.get("welcome",

```

```
[ ]))
```

```
def _spoken(lines) -> str:
    """A door's welcome lines as one spoken sentence; the screen keeps
    them as printed.
```

```

    A line that ends with no punctuation gets a comma, so the voice
    pauses
    where the screen breaks, and Q/A loses its slash. Read 2026-10-02
    in a
    sandbox through the espeak phonemizer inside piper-tts 1.6.0, the
    version
    requirements.txt pins, and not heard on any computer here: Q/A
    comes out
    as "queue slash uh", and QA as Q, A.
    """
    last = len(lines) - 1
    text = " ".join(line if i == last or line[-1:] in ".!?,;:" else
line + ",",
                    for i, line in enumerate(lines))
    return re.sub(r"\b([A-Z])\b", r"\1\2", text)
```

```
def _mark_door_welcomed(path=None):
    """Append welcomed=<date> to .door. Returns None, or the
    OSError."""
    import datetime
    target = Path(path or DOOR_FILE)
    try:
        text = target.read_text(encoding="utf-8")
        lead = "" if not text or text.endswith("\n") else "\n"
        with target.open("a", encoding="utf-8") as handle:
            handle.write(lead + "# Written by first_contact() the day
this door's welcome was shown. "
                        "Delete the welcomed line to see it again.\n"
                        "welcomed=" +
datetime.date.today().isoformat() + "\n")
    except OSError as exc:
        return exc
    return None
```

```
def _show_welcome(door: Dict[str, Any], speak: bool) -> None:
    """Print the door's welcome lines, speak them only if speak, then
    mark .door.
```

```

    Nothing happens without a .door, or once welcomed= is in it. The
    voice
    loads before the lines print, so a 60 MB line, when there is one,
    comes
```

```

    first, and the words on screen arrive with the words spoken. A
door with
    no welcome lines is marked all the same, so the flake's gate shuts
for it.
    """
    if not door or door.get("welcomed"):
        return
    lines = list(door.get("welcome", []))
    voice = chip_voice_system if (speak and lines) else None
    unheard = None
    if voice is not None and not voice.ensure_voice():
        unheard = voice.last_error or "the voice did not load"
        voice = None
    for line in lines:
        print(line, flush=True)
    if voice is not None:
        result = voice.speak_text(_spoken(lines))
        if not result.get("success") and not result.get("declined"):
            unheard = result.get("error") or "nothing was heard"
    if unheard:
        print(f"(the welcome was printed but not heard:
{str(unheard).splitlines()[0][:120]})")
        error = _mark_door_welcomed()
        if error is not None:
            print(f"(could not mark the welcome as shown, so it shows
again next time: {error})")
[[[REPLACE]]]
Target: imports/voice_synthesis.py
[[[SEARCH]]]
    takes ceremony=True from here and from nowhere else. Returns env,
recorded, unavailable, silent, interrupted, yes or no.
    """
    if os.environ.get("PIPULATE_VOICE", "").strip():
        return "env"
    if voice_consent() != "unset":
        return "recorded"
    if not (sys.stdin.isatty() and sys.stdout.isatty()):
        return "unavailable"
    logger.setLevel(logging.CRITICAL)
    voice = chip_voice_system
    try:
        print(FIRST_CONTACT_INTRO, flush=True)
        played = False
        if voice is not None and voice.ensure_voice():
            print(FIRST_CONTACT_LINE, flush=True)
            played = voice.synthesize_and_play(FIRST_CONTACT_LINE,
ceremony=True)
        if not played:

```

```

        reason = (voice.last_error if voice is not None else None)
or "the voice did not load"
        reason = reason.splitlines()[0][:120]
        error = _record_voice_answer("no")
        print(f"The voice could not play here ({reason}), so it
stays off. Type talk to try again.")
        if error is not None:
            print(f"(could not record that, so this runs again
next time: {error})")
        return "silent"
[[[DIVIDER]]]
    takes ceremony=True from here and from nowhere else. Returns env,
    recorded, unavailable, silent, interrupted, yes, no or welcomed.

    AMENDED 2026-10-02 (THE DOOR SPEAKS): an install made through a
door has
    a .door, and the card says that door's words (door_card, above):
its
    workshop line where the intro was, then "This is only
text-to-speech."
    with what a lookup found about local AI, then the same question.
The
    door's welcome lines follow the answer, printed always and spoken
only
    on yes. Where this computer answered the question in an earlier
install,
    flake.nix's gate still opens once for a .door with no welcomed=
line:
    nothing is asked, the workshop line and the welcome print, the
welcome
    is spoken if the recorded answer is yes, welcomed= is written, and
the
    return is welcomed.
"""
if os.environ.get("PIPULATE_VOICE", "").strip():
    return "env"
door = read_door()
recorded = voice_consent()
if recorded != "unset" and (not door or door.get("welcomed")):
    return "recorded"
if not (sys.stdin.isatty() and sys.stdout.isatty()):
    return "unavailable"
logger.setLevel(logging.CRITICAL)
if recorded != "unset":
    if door.get("workshop"):
        print(door["workshop"], flush=True)
    _show_welcome(door, speak=recorded == "yes")
    return "welcomed"

```

```

    voice = chip_voice_system
    intro, ask_lines, _ = door_card(door, local_ai_line() if
door.get("workshop") else "")
    try:
        print(intro, flush=True)
        played = False
        if voice is not None and voice.ensure_voice():
            for line in ask_lines:
                print(line, flush=True)
            played = voice.synthesize_and_play(" ".join(ask_lines),
ceremony=True)
        if not played:
            reason = (voice.last_error if voice is not None else None)
or "the voice did not load"
            reason = reason.splitlines()[0][:120]
            error = _record_voice_answer("no")
            print(f"The voice could not play here ({reason}), so it
stays off. Type talk to try again.")
            if error is not None:
                print(f"(could not record that, so this runs again
next time: {error})")
            _show_welcome(door, speak=False)
            return "silent"
[[[REPLACE]]]
Target: imports/voice_synthesis.py
[[[SEARCH]]]
    if decision == "yes":
        print("Voice on. Type talk any time to turn it off.")
    else:
        print("Staying quiet. Type talk any time to turn the voice
on.")
    return decision
[[[DIVIDER]]]
    if decision == "yes":
        print("Voice on. Type talk any time to turn it off.")
    else:
        print("Staying quiet. Type talk any time to turn the voice
on.")
    _show_welcome(door, speak=decision == "yes")
    return decision
[[[REPLACE]]]
Target: imports/voice_synthesis.py
[[[SEARCH]]]
    # flake.nix's runScript calls this once per machine, right before
the first
    # menu; it exits 0 whatever happened, so the menu always prints
after it.
[[[DIVIDER]]]

```

```

    # flake.nix's runScript calls this once per machine, right before
the first
    # menu, and once more in an install made through a door, for its
welcome;
    # it exits 0 whatever happened, so the menu always prints after
it.
[[[REPLACE]]]

```

Car 2: flake.nix. The welcome gate (ruling 3), the spelled folder (ruling 5), the Lix version line, and the osb word. Expect 4 applied; git counts +41/-3.

Target: flake.nix

```

[[[SEARCH]]]
    NIX_READING=$(LD_LIBRARY_PATH="" nix --version 2>/dev/null |
tail -1 | sed 's/^nix //'')
[[[DIVIDER]]]
    # THE VERSION LINE UNDER LIX (2026-10-02): Lix answers nix
--version
    # with several lines, its version first and a Features line
last, so
    # tail -1 would print the features there. The version is the
line
    # that starts with nix and a space under Nix, Determinate
Nix and
    # Lix alike, so that line is kept and the rest dropped. Read
off a
    # published comparison, not yet witnessed on a Lix machine.
    NIX_READING=$(LD_LIBRARY_PATH="" nix --version 2>/dev/null |
sed -n 's/^nix //p' | head -1)
[[[REPLACE]]]
Target: flake.nix
[[[SEARCH]]]
    # costs a second, never a second question.
    if [ ! -f "$HOME/.config/pipulate/voice" ] && [ -z
'''${PIPULATE_VOICE:-}'" ] && [ -t 0 ] && [ -t 1 ]; then
[[[DIVIDER]]]
    # costs a second, never a second question.
    # AMENDED 2026-10-02 (THE DOOR SPEAKS, the operator's ruling
3: the
    # welcome shows once per install, even on a computer that
answered
    # the question in an earlier install): an install made
through a
    # door carries .door, and until first_contact() writes
welcomed=
    # into it, this gate opens there too. The question is not
asked
    # again; the door's welcome prints, and is spoken if the
recorded

```

```

        # answer is yes. A git clone has no .door, so nothing
changes there.
        VOICE_GATE=shut
        if [ ! -f "$HOME/.config/pipulate/voice" ]; then
VOICE_GATE=open; fi
        if [ -f .door ] && ! grep -q '^welcomed=' .door; then
VOICE_GATE=open; fi
        if [ "$VOICE_GATE" = open ] && [ -z "'${PIPULATE_VOICE:-}"
] && [ -t 0 ] && [ -t 1 ]; then
[[[REPLACE]]]
Target: flake.nix
[[[SEARCH]]]
        TTS_DIR_NAME=$(basename "$PWD")
        TTS_MSG="Pipulate is installed. Starting JupyterLab and
the server. To quit when you are done, forcibly close this window
using $CLOSE_CMD. To run it again later, open a new one, type C, D,
space, $TTS_DIR_NAME, hit enter, then type nix develop, then J, N.
JupyterLab will appear first. Completing the onboarding will unlock
the main application tab. Get ready to hit Shift Enter all the way
down."
[[[DIVIDER]]]
        TTS_DIR_NAME=$(basename "$PWD")
        # THE FOLDER SAID ALOUD (2026-10-02, the operator's ruling
5): Piper
        # reads a folder name as one word, and qamyai comes out
sounding
        # like cam-ee-eye, so the message spells it a letter at a
time with
        # commas between, the way it already says C, D. A dash, an
# underscore and a dot are said as words, because spelled
alone
        # they make no sound. Read through the phonemizer inside
# piper-tts 1.6.0 in a sandbox, not heard on a computer
here.
        TTS_DIR_SPELLED=$(printf '%s' "$TTS_DIR_NAME" | sed
's/./&, /g; s/, $//; s/-/dash/g; s/_/underscore/g; s/[.]/dot/g')
        TTS_MSG="Pipulate is installed. Starting JupyterLab and
the server. To quit when you are done, forcibly close this window
using $CLOSE_CMD. To run it again later, open a new one, type C, D,
space, $TTS_DIR_SPELLED, hit enter, then type nix develop, then J, N.
JupyterLab will appear first. Completing the onboarding will unlock
the main application tab. Get ready to hit Shift Enter all the way
down."
[[[REPLACE]]]
Target: flake.nix
[[[SEARCH]]]
        reseal() { "$PIPULATE_ROOT/.venv/bin/python"
"$PIPULATE_ROOT/imports/ascii_displays.py" --reseal; }

```

```

[[[DIVIDER]]]
    reseal() { "$PIPULATE_ROOT/.venv/bin/python"
"$PIPULATE_ROOT/imports/ascii_displays.py" --reseal; }
    # OPERATION STICK BUG (2026-10-02, the operator's ask: the
letters
    # osb on a menu, a dull name over a loud screen). The one
word that
    # is a blank stare on purpose, THE BLANK STARE RULE's single
    # exception, because looking like a twig is the camouflage.
Typed by
    # a human, so a function. scripts/stick_bug.py takes its
readings
    # at the moment it prints them, then tells a story that says
it is
    # one.
    osb() {
        if [ -f "$PIPULATE_ROOT/scripts/stick_bug.py" ]; then
            "$PIPULATE_ROOT/.venv/bin/python"
"$PIPULATE_ROOT/scripts/stick_bug.py" "$@"
        else
            echo "osb: scripts/stick_bug.py has not landed in this
checkout."
        fi
    }
[[[REPLACE]]]

```

Car 3: scripts/stick_bug.py, the egg. Expect 1 WHOLE-FILE WRITE: CREATED. It's a new 234-line file, so git diff won't list it until it's staged.

Target: scripts/stick_bug.py

```
[[[WRITE_FILE]]]
```

```
#!/usr/bin/env python3
```

```
"""
```

```
stick_bug.py -- osb, OPERATION STICK BUG: the one loud screen in a
quiet workshop.
```

The menu stays dry on purpose; this is what hides under it. A twig that moves, a banner, a block of READINGS this program takes at the moment it prints them, and then six short transmissions that say they are a story. Every reading names what it looked at. Nothing in the story is a reading.

```

    osb                the whole thing; at a keyboard, Enter between
transmissions
    osb --readings     the readings only, then exit (what a compile can
witness)

```

Stdlib first. rich and pyfiglet, which the workshop's venv carries, make it louder; without them it prints the same words in plain text. If narration is on, Piper says one line from a detached child, the way the menu's hint does, and says that it is reading a script.

"""

```
import os
import platform
import re
import sys
import time
from pathlib import Path
```

```
ROOT = Path(__file__).resolve().parent.parent
```

```
TRANSMISSIONS = (
    ("THE TWIG",
     "A stick bug lives by looking like a twig: nothing to see, so
     nothing eats it. "
     "This workshop looks like a dull checklist for checking what an
     AI said, and it "
     "is one. Under the bark there is a clock, a probe, a fence, a
     primer and a "
     "tortoise."),
    ("THE CLOCK",
     "The clock reading above counts seconds from the first second of
     1970, the way "
     "Unix does. Unix was started at Bell Labs in Murray Hill, New
     Jersey, in 1969, "
     "and named as a pun on Multics, a far bigger system: Unics, said
     like eunuchs. "
     "Small, plain and hard to corrupt beat big and clever, and its
     descendants run "
     "under almost everything, this workshop included."),
    ("THE PROBE",
     "John von Neumann worked out, on paper, a machine that builds a
     copy of itself "
     "from parts it finds (Theory of Self-Reproducing Automata, 1966).
     Science fiction "
     "sent such machines to the stars as probes. You typed one line.
     It fetched a "
     "recipe, and the recipe built this workshop from parts on your
     own computer, "
     "pinned to the same versions it pins everywhere. Java promised
     that in 1995: "
```

```

    "write once, run anywhere. Eelco Dolstra's 2006 PhD thesis worked
out how to "
    "deliver it. The thing it describes is Nix."),
    ("THE FENCE",
    "In Michael Crichton's Jurassic Park (1990), the park's computer
counted its "
    "animals by searching for the number it expected, 238, so it
could never find "
    "more. Told to search for more, it kept finding more, 292 in the
end: the "
    "animals were breeding. A check that cannot fail is not a check.
Every check "
    "here is built so that it can fail, and apply.py, the fence
around every edit, "
    "is short enough to read in one sitting. Read it."),
    ("THE PRIMER",
    "In Neal Stephenson's The Diamond Age (1995), a girl named Nell
is raised by a "
    "book that talks back, the Young Lady's Illustrated Primer. It
worked for her "
    "because a person, Miranda, voiced it from far away. The voice in
this workshop "
    "is Piper, a small text-to-speech program, and it says that it is
reading a "
    "script, because a voice should say who is talking."),
    ("THE TORTOISE",
    "In 1895 Lewis Carroll wrote a short dialogue in which a Tortoise
will not accept "
    "a conclusion until Achilles writes the rule that leads to it
down as one more "
    "premise, and then the rule for that one, forever. Douglas
Hofstadter put the "
    "pair back to work in Gödel, Escher, Bach (1979). Asking one AI
to check another "
    "AI's answer is the Tortoise's game: one more premise every time.
The way out is "
    "to run something and read what it printed. That is what a
receipt is."),
)

```

```

FINALE = ("So'wI' yIchu'. (Klingon: engage the cloaking device.)",
          "Type menu to go back to being a twig.")

```

```

TWIG = " " + "=" * 46
BUG = ("  \\      \\      \\",
       " " + "=" * 46 + "<o)",
       " /      /      /")
SPOKEN = "This is Piper, reading a script. You found Operation Stick

```

Bug."

```
def tilde(path: str) -> str:
    home = str(Path.home())
    return "~" + path[len(home):] if path == home or
path.startswith(home + os.sep) else path

def readings() -> list:
    """(label, value) pairs, each taken now; a failed lookup says so
in its row."""
    rows = [("folder", tilde(str(ROOT)))]

    try:
        sys.path.insert(0, str(ROOT))
        from imports.voice_synthesis import read_door, voice_consent,
local_ai_line
        door = read_door().get("door")
        rows.append(("door", f"{door} (read from .door)" if door else
"none (no .door in this folder)"))
        voice = {"yes": "on", "no": "off"}.get(voice_consent(), "not
asked yet")
        local_ai = local_ai_line()
    except Exception as exc:
        unread = f"unread ({type(exc).__name__}: {exc})"[:70]
        rows.append(("door", unread))
        voice = local_ai = unread

    rows.append(("computer", f"{platform.system()}
{platform.machine()}"))

    exe = os.path.realpath(sys.executable)
    if exe.startswith("/nix/store/"):
        store = "/".join(exe.split("/")[:4])
        writable = "writable by you" if os.access(store, os.W_OK) else
"read-only to you"
        rows.append(("python", f"{platform.python_version()} in
{store}"))
        rows.append((" ", f"{writable} (checked just now)"))
    else:
        rows.append(("python", f"{platform.python_version()} at {exe}
(not in the Nix store)"))

    try:
        text = (ROOT / "__init__.py").read_text(encoding="utf-8")
        version = re.search(r'^__version__\s*=\s*"([^"]+)"', text,
re.M).group(1)
        name = re.search(r'^__version_description__\s*=\s*"([^"]+)"',
text, re.M)
```

```

        rows.append(("version", f"v{version}" + (f" {name.group(1)}"
if name else "")))
    except Exception as exc:
        rows.append(("version", f"unread ({type(exc).__name__})"))

    rows.append(("voice", voice))
    rows.append(("local AI", local_ai))
    rows.append(("clock", f"{int(time.time()):,} seconds since 1970
began (Unix time)"))
    return rows

def say(line: str) -> None:
    """Piper says one line if narration is on; never waits, never
raises."""
    if not sys.stdout.isatty():
        return
    try:
        import subprocess
        code = ("import sys; sys.path.insert(0, sys.argv[1]); "
                "from imports.voice_synthesis import chip_voice_system
as v; "
                "v and v.speak_text(sys.argv[2])")
        subprocess.Popen([sys.executable, "-c", code, str(ROOT),
line],
                        stdin=subprocess.DEVNULL,
                        stdout=subprocess.DEVNULL,
                        stderr=subprocess.DEVNULL,
                        start_new_session=True)
    except Exception:
        pass

def print_readings(console=None) -> None:
    title = "OPERATION STICK BUG: readings, each one taken just now"
    rows = readings()
    width = max(len(label) for label, _ in rows)
    if console is not None:
        console.print(f"[bold green]{title}[/bold green]")
        for label, value in rows:
            console.print(f" [green]{label.ljust(width)}[/green]
{value}", highlight=False, soft_wrap=True)
    else:
        print(title)
        for label, value in rows:
            print(f" {label.ljust(width)} {value}")

def wait_or_quit(interactive: bool) -> bool:
    """At a keyboard, Enter goes on and q stops. Returns False to
stop."""

```

```

    if not interactive:
        return True
    try:
        answer = input("    (Enter for the next transmission, q to stop)
")
    except (EOFError, KeyboardInterrupt):
        print()
        return False
    return not answer.strip().lower().startswith("q")

def main() -> int:
    args = set(sys.argv[1:])
    if "--readings" in args:
        print_readings()
        return 0

    interactive = sys.stdin.isatty() and sys.stdout.isatty()
    console = None
    try:
        from rich.console import Console
        console = Console(highlight=False)
    except Exception:
        pass

    print()
    if interactive:
        print(TWIG, flush=True)
        time.sleep(1.2)
        print("\033[1A\033[2K", end="")
    for line in BUG:
        print(line)
    print("        the twig moved.")
    print()
    say(SPOKEN)

    banner = "OPERATION STICK BUG"
    try:
        from pyfiglet import Figlet
        banner = Figlet(font="slant", width=100).renderText("STICK
BUG").rstrip("\n")
    except Exception:
        pass
    if console is not None:
        from rich.panel import Panel
        from rich.text import Text
        console.print(Panel(Text(banner, style="bold green"),
                                title="[bold]osb[/bold]",
                                subtitle="operation stick bug: you typed

```

```

the twig",
                                border_style="green", expand=False))
    else:
        print(banner)
        print("osb -- operation stick bug: you typed the twig")
    print()
    print_readings(console)
    print()

    count = len(TRANSMISSIONS)
    for number, (title, body) in enumerate(TRANSMISSIONS, start=1):
        if not wait_or_quit(interactive):
            break
        heading = f"TRANSMISSION {number} OF {count}: {title}    (a
story; nothing here was measured)"
        if console is not None:
            from rich.panel import Panel
            from rich.text import Text
            console.print(Panel(Text(body), title=f"[bold
green]{heading}[/bold green]",
                                border_style="green",
width=min(console.width, 80)))
        else:
            import textwrap
            print(heading)
            print(textwrap.fill(body, width=76, initial_indent="  ",
subsequent_indent="  "))
            print()
        else:
            for line in FINALE:
                print(line)
    return 0

if __name__ == "__main__":
    sys.exit(main())
[[[END_WRITE_FILE]]]

```

Car 4: GLOSSARY.md. The door entry is marked witnessed and the Stick Bug entry gets osb. Expect 2 applied; git counts +21/-9.

Target: GLOSSARY.md

[[[SEARCH]]]

AMENDED 2026-10-02, THE DOOR NAMES THE ROW (PENDING a witness at a door): the stamp now picks a row in install.sh's door_row, and the row

names the folder, so a door's folder can differ from its stamp: qamy.ai

stamps qamy and installs into ~/qamyai, which said aloud is "QA my AI".

The row also carries the door's words, which install.sh writes to .door

in the new folder: a dotfile, so the first-entry sweep keeps it, and gitignored. The stamped line did not change.

tests/test_install_doors.sh

runs every row with stubs; a real install from qamy.ai landing in ~/qamyai drops PENDING.

[[[DIVIDER]]]

AMENDED 2026-10-02, THE DOOR NAMES THE ROW (WITNESSED at deed 1778, release 2.74 and a Mac install from the door): the stamp now picks a row

in install.sh's door_row, and the row names the folder, so a door's folder can differ from its stamp: qamy.ai stamps qamy and installs into

~/qamyai, which said aloud is "QA my AI". The row also carries the door's

words, which install.sh writes to .door in the new folder: a dotfile, so

the first-entry sweep keeps it, and gitignored. The stamped line did not

change. tests/test_install_doors.sh runs every row with stubs. THE WITNESS: release 2.74's Step 3.1 printed Synced install.sh to npvg.org

and qamy.ai (nixops.sh --installer); the Mac's curl -fsSL https://qamy.ai | bash then printed Qamyai -> ~/qamyai (to remove it later: rm -rf ~/qamyai), and the flake's entry line read

(Determinate

Nix 3.19.1) 2.34.6 · Python 3.12.13 · v2.74 · ~/qamyai. The words rode

along unread at 2.74, so the old card played; THE DOOR SPEAKS is their

reader.

[[[REPLACE]]]

Target: GLOSSARY.md

[[[SEARCH]]]

- **Stick Bug (Operation)** -- *the disappearing instrument.* Inert and

unnoticed until summoned; the human sees a browser and a voice, and the

machinery under them wins by vanishing.

[[[DIVIDER]]]

- **Stick Bug (Operation)** -- *the disappearing instrument.* Inert and

unnoticed until summoned; the human sees a browser and a voice, and the

machinery under them wins by vanishing.

AMENDED 2026-10-02 (PENDING its menu line and a witness): osb is its word, the one abbreviation the workshop prints on purpose, THE BLANK

```
STARE RULE's single exception, because looking like a twig is the
camouflage. Typed, it runs scripts/stick_bug.py, the one loud screen
in
a quiet workshop: readings taken at the moment they print, then six
transmissions that say they are a story.
[[[REPLACE]]]
```

Car 5: foo_files.py. TODOs retired and rewritten, the qamy note fixed, the egg's claim line, and the specimens. Expect 4 applied; git counts +17/-5.

```
Target: foo_files.py
[[[SEARCH]]]
# <-- The one-door command list; short on shell entry, expanded by
`all`, both rendered from tuples in the file
[[[DIVIDER]]]
# <-- The one-door command list; short on shell entry, expanded by
`all`, both rendered from tuples in the file
# scripts/stick_bug.py          # <-- osb, operation stick bug: the egg
under the menu; readings taken as they print, then six transmissions
that say they are a story
[[[REPLACE]]]
Target: foo_files.py
[[[SEARCH]]]
the stamp qamy, so the one-liner lands in ~/qamy
# remotes/honeybot/www/qamy.ai/walk/1/index.html
[[[DIVIDER]]]
the stamp qamy, which picks install.sh's qamy row, so the one-liner
lands in ~/qamyai
# remotes/honeybot/www/qamy.ai/walk/1/index.html
[[[REPLACE]]]
Target: foo_files.py
[[[SEARCH]]]
# - TODO (2026-10-02, THE DOOR SPEAKS, half two; half one is
install.sh's door_row writing .door): first_contact() in
imports/voice_synthesis.py reads .door at the repo root (the module's
parent's parent, never the cwd): key=value lines, # lines skipped,
welcome repeated in order, an empty value treated as absent. It fills
one card: {workshop}; the 60 MB line only when a download happens;
Voice ready.; This is only text-to-speech. {local_ai}; If you can hear
this, press Enter. Type n to keep it quiet.; Enter or n:; Voice on.
Type talk any time to turn it off.; then each {welcome}. local_ai is
No local AI is installed yet. or Local AI detected but not used.,
chosen by code that looks (shutil.which("ollama") at least), because
detected names an act (ATTRIBUTED-VOICE). No .door, or an empty field,
keeps today's words, so a git clone hears no change. Rulings owed: (1)
the operator's target text drops the 60 MB line, kept here as the only
disclosure of a download; (2) welcome lines spoken only when the voice
is on, printed either way; (3) the card is asked once per machine
(~/.config/pipulate/voice), so a second door on one machine never
```

shows its words: per-install welcome or not; (4) installed claims more than a PATH lookup can see, and found is what the code knows; (5) THE SILENT YES stands under the new words. Needs imports/voice_synthesis.py in the payload.

- TODO (2026-10-02, THE PAGE NAMES THE OLD FOLDER): the qamy.ai pages under remotes/honeybot/www may tell a stranger ~/qamy, and install.sh's qamy row now installs ~/qamyai; the ! rg line in the next compile lists every ~/qamy and ~/npvg on the pads. HOLD THE TRUCK (no release, no ./nixops.sh --installer) until this page car and the voice car land, so no door serves a folder its page does not name. The router's own note on remotes/honeybot/www/qamy.ai/index.html (so the one-liner lands in ~/qamy) stays true until the truck runs and changes in the car that carries the truck's witness.

- TODO (2026-10-02, THE FOLDER SAID ALOUD): runScript's TTS_MSG in flake.nix says the folder as one word (\$TTS_DIR_NAME, after type C, D, space), and Piper will not read qamyai as QA my AI; spell it a letter at a time, or carry a spoken= field in .door, ruling owed. The same message opens with Pipulate is installed. on every door, a second place a door's name could ride.

- TODO (2026-10-02, THE VERSION LINE UNDER LIX; read off a published comparison, zenn.dev/trifolium/articles/dalla428c53f65, not witnessed on a machine here): Lix prints four lines for nix --version, the version first and Features: gc, signed-caches last, so runScript's NIX_READING (tail -1, then strip the nix prefix) would print Features: gc, signed-caches on a Lix machine. sed -n 's/^nix //p' | head -1 reads the version line under Nix, Determinate Nix and Lix alike. One line when flake.nix next rides; the witness is the first line a Lix machine prints on entry.

[[[DIVIDER]]]

- TODO (2026-10-02, THE DOOR SPEAKS, LANDED by the reply to deed 1778, WITNESS OWED): first_contact() in imports/voice_synthesis.py reads .door through read_door() and says that door's card (door_card: the workshop line, then This is only text-to-speech. with local_ai_line()'s No local AI found yet. or Local AI detected but not used., then the question); after the answer it prints the welcome lines and speaks them only on yes (_show_welcome; _spoken says Q/A as QA, read through the phonemizer inside piper-tts 1.6.0 in a sandbox), then writes welcomed= into .door. flake.nix's gate also opens for a .door with no welcomed= line, so a second install on a computer that already answered shows its workshop line and welcome once, unasked. With no .door the card is unchanged, byte for byte, in a sandbox run against a fake player. All five rulings landed; ruling 5 is TTS_DIR_SPELLLED in flake.nix. WITNESS OWED, on the Mac: ~/qamyai's second entry after the push (the first one pulls), with ~/.config/pipulate/voice moved aside, shows and says the qamy card and welcome, and a third entry shows neither. The spelled folder is heard on the first jn of an install not yet onboarded. THE SILENT YES stands under the new words.

```

# - TODO (2026-10-02, THE VERSION LINE UNDER LIX, LANDED by the reply
to deed 1778, WITNESS OWED): NIX_READING keeps the line that starts
with nix and a space (sed -n 's/^nix //p' | head -1) where it kept the
last line, read off zenn.dev/trifolium/articles/dall1a428c53f65; Nix
and Determinate Nix print what they printed before, and the witness is
the entry line on the first Lix machine.
# - TODO (2026-10-02, THE EGG ON THE MENU): osb, operation stick bug,
is a word after ignition (flake.nix's osb function runs
scripts/stick_bug.py), and its menu line is owed: ("osb", "operation
stick bug.") as the last row of ALL_WORDS in scripts/boot_menu.py, so
all lists it and the short menu never does. boot_menu.py did not ride
deed 1778, and a car patches only what rode (checklist item 11), so
the line waits for the compile that carries it.
# - TODO (2026-10-02, ONE GRAMMAR FOR THE SLASH, unasked): _spoken in
imports/voice_synthesis.py turns Q/A into QA for a door's welcome
only, because espeak says the slash; the day another spoken line
carries a letter, a slash and a letter, the rule moves into
synthesize_and_play's sanitizer and into
remotes/honeybot/scripts/content_loader.py's clean_markdown in one
commit (TWO LANES, ONE GRAMMAR).
[[[REPLACE]]]
Target: foo_files.py
[[[SEARCH]]]
#   detailed deed 1643 analysis and deliverable documentation" for two
#   files, both named, true.
[[[DIVIDER]]]
#   detailed deed 1643 analysis and deliverable documentation" for two
#   files, both named, true.
#   SPECIMENS 2026-10-01, the door ride (pipulate, same m alias):
43006cba
#   "chore: Update test_install_doors.sh for door table testing" for
three
#   files (+181/-4) that CREATED the test, the door table in
install.sh
#   and the .gitignore entry unnamed; 3f09904d, a GLOSSARY.md-only
+38/-3,
#   got a two-sentence subject whose list ends on "the purpose of the
#   dotfile-blind probe", the entry under the insertion read as the
change
#   (the 0b472c12 shape); 46118ec0 "chore: Remove unused test
#   script" for a foo_files.py-only +6/-1 that CLAIMED the new test
and
#   added five TODOs, the sign read backwards; 34bbaa4b, the 2.74
bump,
#   true; 30dcbaec "chore: Update journal index with new references",
true
#   for one file of two, pyproject.toml's 2.73 to 2.74 unnamed.
[[[REPLACE]]]

```

Ignition, at (nix) pipulate \$: exit, then nix develop, then type osb.

You will see the twig turn into a bug, a green STICK BUG banner, and the nine readings, then (Enter for the next transmission, q to stop). Pressing Enter walks through six transmissions and ends on Type menu to go back to being a twig. With narration on, Piper says "This is Piper, reading a script. You found Operation Stick Bug."

(4) PROMPT

Read this compile's AFTER receipts against your last reply's forecasts: the card probe prints its four lines and rc=0; the egg probe prints its title and nine rows, door none, rc=0; the census prints six lines at 977, 1056, 1062, 1143, 1144 and 1641, rc=0; and Coverage reads 75.7% (212/280). My Mac witness is pasted at the bottom.

1. If the paste shows the qamy card, the two welcome lines and the welcomed= line, mark THE DOOR SPEAKS witnessed in foo_files.py and add the Mac receipt to THE DOOR NAMES THE ROW entry in GLOSSARY.md. If any of the three is missing, name which one and find the cause before patching anything.

2. Put osb on the menu: ("osb", "operation stick bug.") as the last row of ALL_WORDS in scripts/boot_menu.py, which rides this compile, and retire THE EGG ON THE MENU.

3. My rulings:

THE SILENT YES (after the question plays, print and never speak: Heard nothing? Type n.): ____

THE BANNER AND THE LABEL (a door's name on its install banner and app label, through a display-name field in door_row): ____

Mac witness paste:

(5) EXTERNAL DELIVERABLES

Mac witness, after your push.

Step 1. In the Mac's Terminal at michaellevin@MichaelMacBook-Pro ~ %, copy the block below, paste it, and press Enter.

```
cd ~/qamyai && nix develop
```

You will see the usual entry, with update lines while it pulls this turn's commits, then the menu. Nothing new shows yet, because this entry still runs the old gate. Type exit and press Enter.

Step 2. At michaellevin@MichaelMacBook-Pro qamyai %, copy the block below, paste it, and press Enter.

```
mv ~/.config/pipulate/voice ~/.config/pipulate/voice.was && nix develop
```

You will see this before the menu, and hear its last two lines spoken:

```
This LLM Quality Assurance workshop talks you through steps.  
This is only text-to-speech. No local AI found yet.  
If you can hear this, press Enter. Type n to keep it quiet.  
Enter or n:
```

If Ollama is on the Mac, the second line reads Local AI detected but not used.

Press Enter. You will see Voice on. Type talk any time to turn it off., then these two lines, and hear them with QA said as two letters:

```
Welcome. Ready to Q/A your AI with the Flight Data Recorder (FDR)  
Different from Cockpit Voice Recorder (CVR) or "what Claude said".
```

Step 3. At (nix:nix-shell-env) (nix) qamyai \$, copy the block below, paste it, and press Enter.

```
tail -2 .door
```

You will see this, with today's date:

```
# Written by first_contact() the day this door's welcome was shown.  
Delete the welcomed line to see it again.  
welcomed=2026-10-02
```

Step 4. Type exit and press Enter. Then, at michaellevin@MichaelMacBook-Pro qamyai %, copy the block below, paste it, and press Enter.

```
nix develop
```

You will see the menu with no card and no welcome above it. Paste the screens from Steps 2 through 4 into the caboose.

MikeLev.in:

THE AI-EDIT METHOD

Same commands, run twice, one change between them. Where the readings differ is what the change did; the diff in the middle is the receipt.

1: BEFORE:

```
On branch main
```

```
Your branch is up to date with 'origin/main'.
```

```
nothing to commit, working tree clean
```

GIT repo clean. Take BEFORE reading, make CHANGE, record AFTER diff.

```
(nix) pipulate $ cd "$PIPULATE_ROOT" && pwd  
python -c "import tempfile, pathlib; from imports.voice_synthesis  
import read_door, door_card, _spoken, _mark_door_welcomed; t =  
tempfile.TemporaryDirectory(); p = pathlib.Path(t.name) / '.door';  
p.write_text('# c\ndoor=qamy\nworkshop=W.\nwelcome=Ready to Q/A your  
AI\nwelcome=\nwelcome=two\n'); d = read_door(p); print(door_card(d,  
'No local AI found yet.')); print(door_card({})[1]);
```

```

print(_spoken(d['welcome'])); _mark_door_welcomed(p);
print(sorted(read_door(p))); t.cleanup(); echo "rc=$?"
python scripts/stick_bug.py --readings; echo "rc=$?"
rg -n -F -e TTS_DIR_SPELLLED -e 'welcomed=' -e 's/^nix //p' -e 'osb()
{' flake.nix; echo "rc=$?"
/home/mike/repos/pipulate
Traceback (most recent call last):
  File "<string>", line 1, in <module>
ImportError: cannot import name 'read_door' from
'imports.voice_synthesis'
(/home/mike/repos/pipulate/imports/voice_synthesis.py)
rc=1
python: can't open file
'/home/mike/repos/pipulate/scripts/stick_bug.py': [Errno 2] No such
file or directory
rc=2
rc=1
(nix) pipulate $

```

2: AFTER:

```

# # Context 1
# # --- THE 40K-FOOT VIEW (uncomment on a first turn; comment out on
the second) ---
# ! python scripts/articles/lisa.py -t 1 --reverse --fmt dated-slugs #
<-- the book's spine, one line per article, newest first
# ~/repos/nixos/autognome.py # <-- the machine's morning routine
(this author's NixOS box only)
# init.lua # <-- the editor keys that drive the day
# assets/installer/install.sh # <-- how a stranger's machine gets this
workshop
# GLOSSARY.md # <-- the terms, defined
# flake.nix # <-- the environment, pinned: here is
my hardware, here is my state
# prompt_foo.py # <-- the compiler that builds the
payload
# foo_files.py # <-- the router: which files ride, and
this book's outline
# scripts/articles/lisa.py # <-- the second brain: the article
corpus behind `rgx`, `rgxc` and `posts`
# requirements.in # <-- the Python packages, pinned
# pyproject.toml # <-- the PyPI package
# __init__.py # <-- the version

# # Context 2
# ! cd "$PIPULATE_ROOT" && pwd
# ! bash -n assets/installer/install.sh && echo INSTALL_SYNTAX_OK
# ! if command -v nix >/dev/null; then for d in npvg qamy; do
H=$(mktemp -d); mkdir "$H/$d" "$H/${d}ai"; sed

```

```

"s/___INSTALL_DEFAULT_NAME___/$d/g" assets/installer/install.sh |
HOME="$H" bash 2>/dev/null | head -1; rm -rf "$H"; done; else echo
SKIPPED_NO_NIX; fi
# ! bash tests/test_install_doors.sh; echo "rc=$?"
# ! rg -c --hidden --glob '!.git' -e '\.door\b' . ; echo "rc=$?"
# ! rg -n -e '~/(qamy|npvg)\b' -e 'cd (qamy|npvg)\b'
remotes/honeybot/www ; echo "rc=$?"
# imports/voice_synthesis.py
# flake.nix
# assets/installer/install.sh
# tests/test_install_doors.sh
# remotes/honeybot/www/qamy.ai/index.html
# foo_files.py
# GLOSSARY.md

# Context 3
! cd "$PIPULATE_ROOT" && pwd
! python -c "import tempfile, pathlib; from imports.voice_synthesis
import read_door, door_card, _spoken, _mark_door_welcomed; t =
tempfile.TemporaryDirectory(); p = pathlib.Path(t.name) / '.door';
p.write_text('# c\ndoor=qamy\ncworkshop=W.\nwelcome=Ready to Q/A your
AI\nwelcome=\nwelcome=two\n'); d = read_door(p); print(door_card(d,
'No local AI found yet.)); print(door_card({})[1]);
print(_spoken(d['welcome'])); _mark_door_welcomed(p);
print(sorted(read_door(p))); t.cleanup()"; echo "rc=$?"
! python scripts/stick_bug.py --readings; echo "rc=$?"
! rg -n -F -e TTS_DIR_SPELLLED -e 'welcomed=' -e 's/^\nix //p' -e 'osb()
{' flake.nix; echo "rc=$?"
imports/voice_synthesis.py
flake.nix
scripts/stick_bug.py
scripts/boot_menu.py
GLOSSARY.md
foo_files.py

```

3: CHANGE:

On branch main

Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

GIT repo clean. Take BEFORE reading, make CHANGE, record AFTER diff.

```
(nix) pipulate $ patch
```

```
(nix) pipulate $ app
```

```

✓ DETERMINISTIC PATCH APPLIED: Successfully mutated
'imports/voice_synthesis.py'.

```

```

✓ DETERMINISTIC PATCH APPLIED: Successfully mutated
'imports/voice_synthesis.py'.

```

```

✓ DETERMINISTIC PATCH APPLIED: Successfully mutated
'imports/voice_synthesis.py'.
✓ DETERMINISTIC PATCH APPLIED: Successfully mutated
'imports/voice_synthesis.py'.
✓ DETERMINISTIC PATCH APPLIED: Successfully mutated
'imports/voice_synthesis.py'.
(nix) pipulate $ d
diff --git a/imports/voice_synthesis.py b/imports/voice_synthesis.py
index ee89ba99..f63f6d76 100644
--- a/imports/voice_synthesis.py
+++ b/imports/voice_synthesis.py
@@ -11,6 +11,8 @@ import os
     import wave
     import subprocess
     import tempfile
+import re
+import shutil
     import signal
     import sys
     from pathlib import Path
@@ -244,6 +246,136 @@ FIRST_CONTACT_INTRO = ("This workshop can read
its steps aloud with Piper, "
    FIRST_CONTACT_LINE = ("This is Piper, reading a script. If you can
hear this, "

                        "press Enter. Type n to keep it quiet.")

+# THE DOOR SPEAKS (2026-10-02, half two; half one is install.sh's
door_row,
+# which writes .door into every install made through a door).
first_contact()
+# reads .door and says that door's words. With no .door, as in a git
clone,
+# nothing below is used and the card is the one above, unchanged. The
+# operator's five rulings of 2026-10-02, and where each one lives:
+# 1. the 60 MB line stays, printed only when a download happens:
that is
+#     setup_voice_model's own line already, so nothing here repeats
it;
+# 2. the welcome lines print either way and are spoken only when
the
+#     voice is on: _show_welcome;
+# 3. the welcome shows once per install, even on a computer that
answered
+#     the voice question in an earlier install: _mark_door_welcomed
writes
+#     welcomed= into .door, and flake.nix's gate opens until it is
there;
+# 4. "No local AI found yet." rather than "installed", because

```

```

found is
+#      what a lookup knows: local_ai_line;
+#  5. the folder spelled a letter at a time in the spoken install
message:
+#      flake.nix, TTS_DIR_SPELLLED.
+DOOR_FILE = Path(__file__).resolve().parent.parent / ".door"
+DOOR_ASK_LINE = "If you can hear this, press Enter. Type n to keep it
quiet."
+
+
+def read_door(path=None) -> Dict[str, Any]:
+    """The .door at the repo root as a dict, or {} when there is
none.
+
+    One key=value per line, the way install.sh writes it. Lines
starting
+    with # are skipped and an empty value counts as absent. welcome
repeats,
+    so it is a list in file order; any other key keeps its last
value.
+    """
+    door: Dict[str, Any] = {}
+    try:
+        text = Path(path or DOOR_FILE).read_text(encoding="utf-8")
+    except OSError:
+        return door
+    for line in text.splitlines():
+        if line.lstrip().startswith("#") or "=" not in line:
+            continue
+        key, value = (part.strip() for part in line.split("=", 1))
+        if not key or not value:
+            continue
+        if key == "welcome":
+            door.setdefault("welcome", []).append(value)
+        else:
+            door[key] = value
+    return door
+
+
+def local_ai_line() -> str:
+    """One sentence about local AI that claims only what this lookup
found.
+
+    ATTRIBUTED-VOICE: detected names an act, so the act is here, and
it is
+    only a lookup: an ollama on PATH, or the Ollama app in
/Applications.
+    Nothing is started, asked or sent.

```

```

+     """
+     if shutil.which("ollama") or
Path("/Applications/Ollama.app").exists():
+         return "Local AI detected but not used."
+     return "No local AI found yet."
+
+
+def door_card(door: Dict[str, Any], local_ai: str = "") -> tuple:
+     """(intro, ask lines, welcome lines) for first contact. Pure: no
file, no sound.
+
+     A door with no workshop line, or no door at all, gets today's
intro and
+     line exactly and no welcome. The ask lines print one per line and
are
+     spoken as one.
+     """
+     workshop = door.get("workshop", "")
+     if not workshop:
+         return FIRST_CONTACT_INTRO, [FIRST_CONTACT_LINE], []
+     first = f"This is only text-to-speech. {local_ai}".strip()
+     return workshop, [first, DOOR_ASK_LINE], list(door.get("welcome",
[]))
+
+
+def _spoken(lines) -> str:
+     """A door's welcome lines as one spoken sentence; the screen
keeps them as printed.
+
+     A line that ends with no punctuation gets a comma, so the voice
pauses
+     where the screen breaks, and Q/A loses its slash. Read 2026-10-02
in a
+     sandbox through the espeak phonemizer inside piper-tts 1.6.0, the
version
+     requirements.txt pins, and not heard on any computer here: Q/A
comes out
+     as "queue slash uh", and QA as Q, A.
+     """
+     last = len(lines) - 1
+     text = " ".join(line if i == last or line[-1:] in ".!?,;:" else
line + ",",
+
+                     for i, line in enumerate(lines))
+     return re.sub(r"\b([A-Z])/([A-Z])\b", r"\1\2", text)
+
+
+def _mark_door_welcomed(path=None):
+     """Append welcomed=<date> to .door. Returns None, or the

```

```

OSError."""
+     import datetime
+     target = Path(path or DOOR_FILE)
+     try:
+         text = target.read_text(encoding="utf-8")
+         lead = "" if not text or text.endswith("\n") else "\n"
+         with target.open("a", encoding="utf-8") as handle:
+             handle.write(lead + "# Written by first_contact() the day
this door's welcome was shown. "
+                             "Delete the welcomed line to see it
again.\n"
+                             "welcomed=" +
datetime.date.today().isoformat() + "\n")
+     except OSError as exc:
+         return exc
+     return None
+
+
+def _show_welcome(door: Dict[str, Any], speak: bool) -> None:
+    """Print the door's welcome lines, speak them only if speak, then
mark .door.
+
+    Nothing happens without a .door, or once welcomed= is in it. The
voice
+    loads before the lines print, so a 60 MB line, when there is one,
comes
+    first, and the words on screen arrive with the words spoken. A
door with
+    no welcome lines is marked all the same, so the flake's gate
shuts for it.
+    """
+    if not door or door.get("welcomed"):
+        return
+    lines = list(door.get("welcome", []))
+    voice = chip_voice_system if (speak and lines) else None
+    unheard = None
+    if voice is not None and not voice.ensure_voice():
+        unheard = voice.last_error or "the voice did not load"
+        voice = None
+    for line in lines:
+        print(line, flush=True)
+    if voice is not None:
+        result = voice.speak_text(_spoken(lines))
+        if not result.get("success") and not result.get("declined"):
+            unheard = result.get("error") or "nothing was heard"
+    if unheard:
+        print(f"(the welcome was printed but not heard:
{str(unheard).splitlines()[0][:120]})")

```

```

+     error = _mark_door_welcomed()
+     if error is not None:
+         print(f"(could not mark the welcome as shown, so it shows
again next time: {error})")
+
def first_contact() -> str:
    """Ask by voice, once per machine, right before the first menu.
@@ -259,22 +391,42 @@ def first_contact() -> str:
    nothing, so the question returns next time. The one line spoken
before any
    answer exists is the question itself, which is why
synthesize_and_play
    takes ceremony=True from here and from nowhere else. Returns env,
-    recorded, unavailable, silent, interrupted, yes or no.
+    recorded, unavailable, silent, interrupted, yes, no or welcomed.
+
+    AMENDED 2026-10-02 (THE DOOR SPEAKS): an install made through a
door has
+    a .door, and the card says that door's words (door_card, above):
its
+    workshop line where the intro was, then "This is only
text-to-speech."
+    with what a lookup found about local AI, then the same question.
The
+    door's welcome lines follow the answer, printed always and spoken
only
+    on yes. Where this computer answered the question in an earlier
install,
+    flake.nix's gate still opens once for a .door with no welcomed=
line:
+    nothing is asked, the workshop line and the welcome print, the
welcome
+    is spoken if the recorded answer is yes, welcomed= is written,
and the
+    return is welcomed.
    """
    if os.environ.get("PIPULATE_VOICE", "").strip():
        return "env"
-    if voice_consent() != "unset":
+    door = read_door()
+    recorded = voice_consent()
+    if recorded != "unset" and (not door or door.get("welcomed")):
        return "recorded"
    if not (sys.stdin.isatty() and sys.stdout.isatty()):
        return "unavailable"
    logger.setLevel(logging.CRITICAL)
+    if recorded != "unset":

```

```

+         if door.get("workshop"):
+             print(door["workshop"], flush=True)
+             _show_welcome(door, speak=recorded == "yes")
+             return "welcomed"
        voice = chip_voice_system
+        intro, ask_lines, _ = door_card(door, local_ai_line() if
door.get("workshop") else "")
        try:
-            print(FIRST_CONTACT_INTRO, flush=True)
+            print(intro, flush=True)
            played = False
            if voice is not None and voice.ensure_voice():
-                print(FIRST_CONTACT_LINE, flush=True)
-                played = voice.synthesize_and_play(FIRST_CONTACT_LINE,
ceremony=True)
+                for line in ask_lines:
+                    print(line, flush=True)
+                    played = voice.synthesize_and_play(" ".join(ask_lines),
ceremony=True)
            if not played:
                reason = (voice.last_error if voice is not None else
None) or "the voice did not load"
                reason = reason.splitlines()[0][:120]
@@ -282,6 +434,7 @@ def first_contact() -> str:
                print(f"The voice could not play here ({reason}), so it
stays off. Type talk to try again.")
                if error is not None:
                    print(f"(could not record that, so this runs again
next time: {error})")
+                _show_welcome(door, speak=False)
                return "silent"
                with open("/dev/tty", "r", encoding="utf-8") as tty:
                    print("Enter or n: ", end="", flush=True)
@@ -303,6 +456,7 @@ def first_contact() -> str:
                    print("Voice on. Type talk any time to turn it off.")
                else:
                    print("Staying quiet. Type talk any time to turn the voice
on.")
+                _show_welcome(door, speak=decision == "yes")
                return decision

@@ -850,7 +1004,8 @@ if __name__ == "__main__" and sys.argv[1:2] ==
["talk"]:
    raise SystemExit(talk_command(sys.argv[2:]))
    if __name__ == "__main__" and sys.argv[1:2] == ["first"]:
        # flake.nix's runScript calls this once per machine, right before
the first

```

```

-   # menu; it exits 0 whatever happened, so the menu always prints
after it.
+   # menu, and once more in an install made through a door, for its
welcome;
+   # it exits 0 whatever happened, so the menu always prints after
it.
    try:
        first_contact()
    except Exception as _exc:
(nix) pipulate $ m
📝 Committing: feat: add .door read and first_contact() implementation
[main 868425bd] feat: add .door read and first_contact()
implementation
1 file changed, 161 insertions(+), 6 deletions(-)
(nix) pipulate $ patch
(nix) pipulate $ app
✅ DETERMINISTIC PATCH APPLIED: Successfully mutated 'flake.nix'.
✅ DETERMINISTIC PATCH APPLIED: Successfully mutated 'flake.nix'.
✅ DETERMINISTIC PATCH APPLIED: Successfully mutated 'flake.nix'.
✅ DETERMINISTIC PATCH APPLIED: Successfully mutated 'flake.nix'.
(nix) pipulate $ d
diff --git a/flake.nix b/flake.nix
index 3b7731f5..20a4ec72 100644
--- a/flake.nix
+++ b/flake.nix
@@ -968,7 +968,13 @@ runScript = pkgs.writeShellScriptBin "run-script"
''
        # bare dollar followed by a letter is literal and safe,
which is why
        # $PWD and $HOME below reach bash intact. Two adjacent
single quotes
        # are the other structural hazard and appear nowhere here.
-       NIX_READING=$(LD_LIBRARY_PATH="" nix --version 2>/dev/null
| tail -1 | sed 's/^nix //'')
+       # THE VERSION LINE UNDER LIX (2026-10-02): Lix answers nix
--version
+       # with several lines, its version first and a Features line
last, so
+       # tail -1 would print the features there. The version is
the line
+       # that starts with nix and a space under Nix, Determinate
Nix and
+       # Lix alike, so that line is kept and the rest dropped.
Read off a
+       # published comparison, not yet witnessed on a Lix machine.
+       NIX_READING=$(LD_LIBRARY_PATH="" nix --version 2>/dev/null
| sed -n 's/^nix //p' | head -1)
        if [ -z "$NIX_READING" ]; then NIX_READING="nix: no

```

```

reading"; fi
    PY_READING=$(python -V 2>&1)
    if [ -z "$PY_READING" ]; then PY_READING="python: no
reading"; fi
@@ -1044,7 +1050,17 @@ runScript = pkgs.writeShellScriptBin
"run-script" ''
    # later entry. The file test spares later entries an
import; the
    # module rereads the answer and decides, so a drift between
the two
    # costs a second, never a second question.
-    if [ ! -f "$HOME/.config/pipulate/voice" ] && [ -z
"'${PIPULATE_VOICE:-}" ] && [ -t 0 ] && [ -t 1 ]; then
+    # AMENDED 2026-10-02 (THE DOOR SPEAKS, the operator's
ruling 3: the
+    # welcome shows once per install, even on a computer that
answered
+    # the question in an earlier install): an install made
through a
+    # door carries .door, and until first_contact() writes
welcomed=
+    # into it, this gate opens there too. The question is not
asked
+    # again; the door's welcome prints, and is spoken if the
recorded
+    # answer is yes. A git clone has no .door, so nothing
changes there.
+    VOICE_GATE=shut
+    if [ ! -f "$HOME/.config/pipulate/voice" ]; then
VOICE_GATE=open; fi
+    if [ -f .door ] && ! grep -q '^welcomed=' .door; then
VOICE_GATE=open; fi
+    if [ "$VOICE_GATE" = open ] && [ -z "'${PIPULATE_VOICE:-}"
] && [ -t 0 ] && [ -t 1 ]; then
        case "'${PIPULATE_BOOT_MENU:-1}" in
            0|no|off|false) ;;
            *) python imports/voice_synthesis.py first || true ;;
@@ -1117,7 +1133,15 @@ runScript = pkgs.writeShellScriptBin
"run-script" ''
    CLOSE_CMD="clicking the X"
fi
    TTS_DIR_NAME=$(basename "$PWD")
-    TTS_MSG="Pipulate is installed. Starting JupyterLab and
the server. To quit when you are done, forcibly close this window
using $CLOSE_CMD. To run it again later, open a new one, type C, D,
space, $TTS_DIR_NAME, hit enter, then type nix develop, then J, N.
JupyterLab will appear first. Completing the onboarding will unlock
the main application tab. Get ready to hit Shift Enter all the way

```

```

down."
+           # THE FOLDER SAID ALOUD (2026-10-02, the operator's
ruling 5): Piper
+           # reads a folder name as one word, and qamyai comes out
sounding
+           # like cam-ee-eye, so the message spells it a letter at a
time with
+           # commas between, the way it already says C, D. A dash,
an
+           # underscore and a dot are said as words, because spelled
alone
+           # they make no sound. Read through the phonemizer inside
+           # piper-tts 1.6.0 in a sandbox, not heard on a computer
here.
+           TTS_DIR_SPELLED=$(printf '%s' "$TTS_DIR_NAME" | sed
's/./&, /g; s/, $//; s/-/dash/g; s/_/underscore/g; s/[.]/dot/g')
+           TTS_MSG="Pipulate is installed. Starting JupyterLab and
the server. To quit when you are done, forcibly close this window
using $CLOSE_CMD. To run it again later, open a new one, type C, D,
space, $TTS_DIR_SPELLED, hit enter, then type nix develop, then J, N.
JupyterLab will appear first. Completing the onboarding will unlock
the main application tab. Get ready to hit Shift Enter all the way
down."

           python -c "import logging;
logging.getLogger('piper').setLevel(logging.ERROR); from
imports.voice_synthesis import chip_voice_system as cvs;
cvs.speak_text('$TTS_MSG')" > /dev/null 2>&1 &
           fi

@@ -1607,6 +1631,20 @@ runScript = pkgs.writeShellScriptBin
"run-script" ''
           fi
       }
       reseal() { "$PIPULATE_ROOT/.venv/bin/python"
"$PIPULATE_ROOT/imports/ascii_displays.py" --reseal; }
+           # OPERATION STICK BUG (2026-10-02, the operator's ask: the
letters
+           # osb on a menu, a dull name over a loud screen). The one
word that
+           # is a blank stare on purpose, THE BLANK STARE RULE's
single
+           # exception, because looking like a twig is the camouflage.
Typed by
+           # a human, so a function. scripts/stick_bug.py takes its
readings
+           # at the moment it prints them, then tells a story that
says it is
+           # one.

```

```

+         osb() {
+             if [ -f "$PIPULATE_ROOT/scripts/stick_bug.py" ]; then
+                 "$PIPULATE_ROOT/.venv/bin/python"
"$PIPULATE_ROOT/scripts/stick_bug.py" "$@"
+             else
+                 echo "osb: scripts/stick_bug.py has not landed in this
checkout."
+             fi
+         }
+         tools() {
+             if [ "$#" -eq 0 ]; then
+                 (cd "$PIPULATE_ROOT" && .venv/bin/python cli.py
mcp-discover)
(nix) pipulate $ m
📝 Committing: chore: Update Nix version reading and voice welcome
message
[main 869637dc] chore: Update Nix version reading and voice welcome
message
1 file changed, 41 insertions(+), 3 deletions(-)
(nix) pipulate $ patch
(nix) pipulate $ app
✅ WHOLE-FILE WRITE: CREATED 'scripts/stick_bug.py'.
(nix) pipulate $ git add scripts/stick_bug.py
(nix) pipulate $ m
📝 Committing: chore: Update stick_bug.py with more detailed
explanations and improvements
[main daaac215] chore: Update stick_bug.py with more detailed
explanations and improvements
1 file changed, 234 insertions(+)
create mode 100644 scripts/stick_bug.py
(nix) pipulate $ patch
(nix) pipulate $ app
✅ DETERMINISTIC PATCH APPLIED: Successfully mutated 'GLOSSARY.md'.
✅ DETERMINISTIC PATCH APPLIED: Successfully mutated 'GLOSSARY.md'.
(nix) pipulate $ d
diff --git a/GLOSSARY.md b/GLOSSARY.md
index 5ea8f58b..dc062f32 100644
--- a/GLOSSARY.md
+++ b/GLOSSARY.md
@@ -417,6 +417,12 @@ Entries are alphabetical, numbers spelled as
spoken.
- **Stick Bug (Operation)** -- *the disappearing instrument.* Inert
and
unnoticed until summoned; the human sees a browser and a voice, and
the
machinery under them wins by vanishing.
+ AMENDED 2026-10-02 (PENDING its menu line and a witness): osb is
its

```

```

+ word, the one abbreviation the workshop prints on purpose, THE
BLANK
+ STARE RULE's single exception, because looking like a twig is the
+ camouflage. Typed, it runs scripts/stick_bug.py, the one loud
screen in
+ a quiet workshop: readings taken at the moment they print, then six
+ transmissions that say they are a story.
- **Straddle** -- *before/after measurement bracketing execution.*
One probe,
    two receipts, patch (and ignition) between; the binary-search
causal
    boundary that removes ordering ambiguity.
@@ -1003,15 +1009,21 @@ Entries are alphabetical, numbers spelled as
spoken.
    and an nginx restart. A body-only change to a served file is one
rsync;
    nixops.sh prints the rebuild line as an offer, and publish pairs
the two
    because a publish may carry config.
- AMENDED 2026-10-02, THE DOOR NAMES THE ROW (PENDING a witness at a
- door): the stamp now picks a row in install.sh's door_row, and the
row
- names the folder, so a door's folder can differ from its stamp:
qamy.ai
- stamps qamy and installs into ~/qamyai, which said aloud is "QA my
AI".
- The row also carries the door's words, which install.sh writes to
.door
- in the new folder: a dotfile, so the first-entry sweep keeps it,
and
- gitignored. The stamped line did not change.
tests/test_install_doors.sh
- runs every row with stubs; a real install from qamy.ai landing in
- ~/qamyai drops PENDING.
+ AMENDED 2026-10-02, THE DOOR NAMES THE ROW (WITNESSED at deed 1778,
+ release 2.74 and a Mac install from the door): the stamp now picks
a row
+ in install.sh's door_row, and the row names the folder, so a door's
+ folder can differ from its stamp: qamy.ai stamps qamy and installs
into
+ ~/qamyai, which said aloud is "QA my AI". The row also carries the
door's
+ words, which install.sh writes to .door in the new folder: a
dotfile, so
+ the first-entry sweep keeps it, and gitignored. The stamped line
did not
+ change. tests/test_install_doors.sh runs every row with stubs. THE
+ WITNESS: release 2.74's Step 3.1 printed Synced install.sh to

```

```

npvg.org
+ and qamy.ai (nixops.sh --installer); the Mac's curl -fsSL
+ https://qamy.ai | bash then printed Qamyai -> ~/qamyai (to remove
it
+ later: rm -rf ~/qamyai), and the flake's entry line read
(Determinate
+ Nix 3.19.1) 2.34.6 · Python 3.12.13 · v2.74 · ~/qamyai. The words
rode
+ along unread at 2.74, so the old card played; THE DOOR SPEAKS is
their
+ reader.
- **The Dotfile-Blind Probe** -- *ripgrep's hidden-file default.*
Banked
    2026-08-05, self-convicted. ripgrep skips hidden files by default,
so any rg
    probe hunting a name that lives in .gitignore, .gitattributes, or
any other
(nix) pipulate $ m
📝 Committing: chore: Update GLOSSARY.md with detailed witness notes
and clarifications for "Stick Bug" and "The Dotfile-Blind Probe"
[main 8149a74c] chore: Update GLOSSARY.md with detailed witness notes
and clarifications for "Stick Bug" and "The Dotfile-Blind Probe"
    1 file changed, 21 insertions(+), 9 deletions(-)
(nix) pipulate $ patch
(nix) pipulate $ app
✅ DETERMINISTIC PATCH APPLIED: Successfully mutated 'foo_files.py'.
✅ DETERMINISTIC PATCH APPLIED: Successfully mutated 'foo_files.py'.
✅ DETERMINISTIC PATCH APPLIED: Successfully mutated 'foo_files.py'.
✅ DETERMINISTIC PATCH APPLIED: Successfully mutated 'foo_files.py'.
(nix) pipulate $ d
diff --git a/foo_files.py b/foo_files.py
index 8ba9fb4b..ab80accb 100644
--- a/foo_files.py
+++ b/foo_files.py
@@ -1460,6 +1460,7 @@ foo_files.py      # <-- THIS file. Content
compiler router. Makes it very meta.
# assets/installer/mck.sh      # <-- THE LAUNCHER (curl -fsSL
pipulate.com/mck.sh | bash): finds the workshop, forces the spoken
rehearsal on first contact, asks for the word RIDE
# scripts/mother_cat.py      # <-- THE RIDER (alias: mothercat), Car
B: actuates walk.py's validated plan; per-run capture banking, DECANT,
the context.txt writer
# scripts/boot_menu.py      # <-- The one-door command list; short
on shell entry, expanded by `all`, both rendered from tuples in the
file
+# scripts/stick_bug.py      # <-- osb, operation stick bug: the egg
under the menu; readings taken as they print, then six transmissions
that say they are a story

```

```

# scripts/sources_menu.py      # <-- The roster `connect` prints;
filename stays descriptive, conn is the short spelling
#
# AUTHOR
@@ -1502,7 +1503,7 @@ foo_files.py      # <-- THIS file. Content
compiler router. Makes it very meta.
# remotes/honeybot/www/npvg.org/walk/3/index.html  # <-- stop three:
the walk's only script rewrites the server's sentence and appends a
paragraph; the DECANT test lives here
#
# THE SECOND DOOR (qamy.ai; landed 2026-09-29): npvg.org's tree
copied, its own vhost, certificate and log in configuration.nix, four
lines in nixops.sh; only the two command lines differ until the
divergence ride, and no trail names these stops yet
-# remotes/honeybot/www/qamy.ai/index.html          # <-- the door:
the same negotiation as npvg.org's, the stamp qamy, so the one-liner
lands in ~/qamy
+# remotes/honeybot/www/qamy.ai/index.html          # <-- the door:
the same negotiation as npvg.org's, the stamp qamy, which picks
install.sh's qamy row, so the one-liner lands in ~/qamyai
# remotes/honeybot/www/qamy.ai/walk/1/index.html    # <-- npvg.org's
stop one, copied
# remotes/honeybot/www/qamy.ai/walk/2/index.html    # <-- stop two,
copied; its ls path still spells npvg.org
# remotes/honeybot/www/qamy.ai/walk/3/index.html    # <-- stop three,
copied
@@ -2358,10 +2359,10 @@ MATCHBOOK_CHOP = r"""
# - TODO (2026-10-01, THE UPDATE PATH SAYS WHAT IT DOES NOT DO; read
off flake.nix while charging the operator's Useless Machine its rent,
every branch INFERRED until a manufactured state prints it):
gitUpdateLogic compares HEAD and origin/main with != alone, so a
checkout AHEAD of origin/main prints Updates found, then git's Already
up to date., then Update complete!; a dirty tree prints Skipping
automatic update, and then runScript's bare git pull --quiet (it runs
in every shell that runs runScript) pulls anyway when the incoming
commits miss the dirty files, git refusing only an overlap, so no work
is lost and only the sentence is false; the dev shell prints skipping
automatic git update and then runs that same pull. The DIVERGED branch
is correct and is the operator's Useless Machine: the updater's only
act is refusing, every entry, until a human runs one of its two
printed commands. Cure, one car when flake.nix rides: git merge-base
--is-ancestor splits != into behind (pull), ahead (one line with git
rev-list --count origin/main..HEAD: push them so your other machines
get them) and diverged (refuse); runScript's pull runs only where the
dev shell asks for it, and the dev hook says what dev does, the
operator ruling whether dev pulls. Gate: git commit --allow-empty -m
probe, exit, nix develop reads the ahead line and not Updates found;
git reset --hard HEAD~1 undoes it.

```

- TODO (2026-10-01, THE SILENT YES; operator's ruling owed):
first_contact() reads Enter alone as yes, so its hearing test passes on silence: a player exiting 0 proves the sound left the program, never that it reached an ear (THE SUCCESS-ONLY WITNESS, aimed at a speaker), and If you can hear this, press Enter says nothing to someone who heard nothing. Cheapest cure, unasked: one line printed after playback and never spoken, Heard nothing? Type n. Cost of a false yes today: talk turns it off, and INFERRED that a walk waits through each spoken stop (synthesize_and_play waits on the player; mother_cat.py's call is unread). Gate: the ruling, then one first entry with the volume at zero.

- TODO (2026-10-01, THE VOICE ASKS ONCE, OUT LOUD; landed by deed 1748's Car 1): first_contact() in imports/voice_synthesis.py, called by runScript just before the menu where no answer is recorded, at a keyboard, PIPULATE_VOICE unset and the menu not declined: one printed line says what Piper is, setup_voice_model prints its own 60 MB line before it fetches an absent model, the voice speaks one scripted line asking the question (printed at the same moment), Enter keeps it and n turns it off; a voice that cannot load or play records no and says why in one line; Ctrl+C or an empty read records nothing; mother_cat.py:128 asks without force, so the walk's card finds the answer and stays quiet. The compile lane has no keyboard and no ears, so its probe reads only the silent world (0 lines); the rest is witnessed by hand or not at all. UNWITNESSED: the download branch (Prime and the Mac hold the model; assets/piper_models moved aside, or a fresh install, reads it), the blocked-download branch (the model aside and HF_ENDPOINT pointed at a closed local port is the cheap fake, INFERRED that the hub reads HF_ENDPOINT at import), a machine with no sound, the Mac, and the walk lane after mck.sh's install-only path, which never enters the default room and so still meets the card. Gate: a pasted first entry shows the intro line, the question, the answer line and the menu, and the voice file's date moved.

-# - TODO (2026-10-02, THE DOOR SPEAKS, half two; half one is install.sh's door_row writing .door): first_contact() in imports/voice_synthesis.py reads .door at the repo root (the module's parent's parent, never the cwd): key=value lines, # lines skipped, welcome repeated in order, an empty value treated as absent. It fills one card: {workshop}; the 60 MB line only when a download happens; Voice ready.; This is only text-to-speech. {local_ai}; If you can hear this, press Enter. Type n to keep it quiet.; Enter or n:; Voice on. Type talk any time to turn it off.; then each {welcome}. local_ai is No local AI is installed yet. or Local AI detected but not used., chosen by code that looks (shutil.which("ollama") at least), because detected names an act (ATTRIBUTED-VOICE). No .door, or an empty field, keeps today's words, so a git clone hears no change. Rulings owed: (1) the operator's target text drops the 60 MB line, kept here as the only disclosure of a download; (2) welcome lines spoken only when the voice is on, printed either way; (3) the card is asked once per machine

(~/config/pipulate/voice), so a second door on one machine never shows its words: per-install welcome or not; (4) installed claims more than a PATH lookup can see, and found is what the code knows; (5) THE SILENT YES stands under the new words. Needs imports/voice_synthesis.py in the payload.

-# - TODO (2026-10-02, THE PAGE NAMES THE OLD FOLDER): the qamy.ai pages under remotes/honeybot/www may tell a stranger ~/qamy, and install.sh's qamy row now installs ~/qamyai; the ! rg line in the next compile lists every ~/qamy and ~/npvg on the pads. HOLD THE TRUCK (no release, no ./nixops.sh --installer) until this page car and the voice car land, so no door serves a folder its page does not name. The router's own note on remotes/honeybot/www/qamy.ai/index.html (so the one-liner lands in ~/qamy) stays true until the truck runs and changes in the car that carries the truck's witness.

-# - TODO (2026-10-02, THE FOLDER SAID ALOUD): runScript's TTS_MSG in flake.nix says the folder as one word (\$TTS_DIR_NAME, after type C, D, space), and Piper will not read qamyai as QA my AI; spell it a letter at a time, or carry a spoken= field in .door, ruling owed. The same message opens with Pipulate is installed. on every door, a second place a door's name could ride.

-# - TODO (2026-10-02, THE VERSION LINE UNDER LIX; read off a published comparison, zenn.dev/trifolium/articles/dalla428c53f65, not witnessed on a machine here): Lix prints four lines for nix --version, the version first and Features: gc, signed-caches last, so runScript's NIX_READING (tail -1, then strip the nix prefix) would print Features: gc, signed-caches on a Lix machine. sed -n 's/^nix //p' | head -1 reads the version line under Nix, Determinate Nix and Lix alike. One line when flake.nix next rides; the witness is the first line a Lix machine prints on entry.

+# - TODO (2026-10-02, THE DOOR SPEAKS, LANDED by the reply to deed 1778, WITNESS OWED): first_contact() in imports/voice_synthesis.py reads .door through read_door() and says that door's card (door_card: the workshop line, then This is only text-to-speech. with local_ai_line()'s No local AI found yet. or Local AI detected but not used., then the question); after the answer it prints the welcome lines and speaks them only on yes (_show_welcome; _spoken says Q/A as QA, read through the phonemizer inside piper-tts 1.6.0 in a sandbox), then writes welcomed= into .door. flake.nix's gate also opens for a .door with no welcomed= line, so a second install on a computer that already answered shows its workshop line and welcome once, unasked. With no .door the card is unchanged, byte for byte, in a sandbox run against a fake player. All five rulings landed; ruling 5 is TTS_DIR_SPELLLED in flake.nix. WITNESS OWED, on the Mac: ~/qamyai's second entry after the push (the first one pulls), with ~/.config/pipulate/voice moved aside, shows and says the qamy card and welcome, and a third entry shows neither. The spelled folder is heard on the first jn of an install not yet onboarded. THE SILENT YES stands under the new words.

+# - TODO (2026-10-02, THE VERSION LINE UNDER LIX, LANDED by the reply to deed 1778, WITNESS OWED): NIX_READING keeps the line that starts with nix and a space (sed -n 's/^nix //p' | head -1) where it kept the last line, read off zenn.dev/trifolium/articles/dall1a428c53f65; Nix and Determinate Nix print what they printed before, and the witness is the entry line on the first Lix machine.

+# - TODO (2026-10-02, THE EGG ON THE MENU): osb, operation stick bug, is a word after ignition (flake.nix's osb function runs scripts/stick_bug.py), and its menu line is owed: ("osb", "operation stick bug.") as the last row of ALL_WORDS in scripts/boot_menu.py, so all lists it and the short menu never does. boot_menu.py did not ride deed 1778, and a car patches only what rode (checklist item 11), so the line waits for the compile that carries it.

+# - TODO (2026-10-02, ONE GRAMMAR FOR THE SLASH, unasked): _spoken in imports/voice_synthesis.py turns Q/A into QA for a door's welcome only, because espeak says the slash; the day another spoken line carries a letter, a slash and a letter, the rule moves into synthesize_and_play's sanitizer and into remotes/honeybot/scripts/content_loader.py's clean_markdown in one commit (TWO LANES, ONE GRAMMAR).

- TODO (2026-10-02, THE BANNER AND THE LABEL, unasked): install.sh's first line capitalizes the folder (Qamyai -> ~/qamyai), and a default install from any door is named Pipulate by the flake, because whitelabel.txt is written only for an argument; a display-name field in door_row would carry QA my AI to both places. Ruling owed: should a door's app carry the door's name.

- TODO (2026-09-30, THE TALK FLIP): talk rides the menu under walk as a second spelling of voice (the ask card, y or n); the flip it was asked to be needs imports/voice_synthesis.py in the payload, because that file owns the recorded answer, its format and the barrier that reads it. Shape, unriden: a flip branch beside ask; bare talk flips and prints narration: on or narration: off, speaking one sentence on on; talk on and talk off set it, the first positional carrying the state; nothing recorded runs the full card instead, because the card carries the download disclosure (CONSENT CANNOT PRECEDE ITS OBJECT); PIPULATE_VOICE set in the shell is named when it outranks the file; then the card and mother_cat.py's later_hint name talk. Gate: two bare talks print opposite states, and the recorded answer between them reads the first. LANDED 2026-09-30 (deed 1740's car): the format read off the file is one word and a newline (VOICE_ANSWER regular file 4 bytes, yes), written by the card alone until now and read fresh by voice_consent() on every call, so a flip holds at the next spoken line in every process; talk_command sits beside ask with _record_voice_answer as the one writer the card now shares; with nothing recorded every spelling runs the card, so talk writes only over an answer the card recorded; PIPULATE_VOICE outranks the file only as one of the eight words voice_consent() reads, and talk then names it and writes nothing, while the voice word still refuses on any

non-empty value, a difference left standing; the card's hint and mother_cat.py's later_hint name talk. The gate is Car 3's hand line; the scratch probe reads the same branches with HOME in a temp folder and the speaker unplugged.

- TODO (2026-09-29, THE POST IS THE DELIVERABLE; the operator's ruling at deed 1695: "If it doesn't end up in that zip archive at the end then it doesn't exist", "I type the walk command. The walk happens. A context.txt gets written to. I compile it. I have an archive. End of story."): deliverable.md is retired. THE ANSWER IS THE ARTIFACT already said so (the sealed zip and the copyable text of the reply are the only durable outputs) and the file broke it the day it was born, landing in the corporate tier by a car the operator had to git add by hand. From here the deliverable is the reply's first fence, the Slack post in SVB_PROMPT.md's shape; the operator pastes the reply into the article as every turn, the next compile seals it in prompt.md, the zip goes to Drive and the post to Slack with the link in its one slot (the zip's filename is the deed, so the deed parenthetical leaves the Archive line). SVB_PROMPT.md is the SVB article's dismount protocol; introduction.md stays the walk's newcomer page. Three files change on the next compile, one car each, none in deed 1695's payload: CONTEXT_TEMPLATE.txt loses its THE DELIVERABLE block and the {DELIVERABLE} token; svb.py loses the file check, the header's "deliverable not yet written" and the # Next: hint naming --prompt; SVB_PROMPT.md loses step 3 and the Archive line's deed clause. SVB-115's deliverable.md stays as the specimen's history; SVB-133's left at deed 1695's Car 2, its text sealed in that deed's payload. The 2026-09-27 deliverable.html ruling (a fourth member) and the house-subset renderer are superseded: three members, no file. Gate: svb KEY --write lays a board with no deliverable line, and the second ticket's post reaches Slack from the article alone. READ 2026-09-29 (deed 1696, the dismount): the three cars landed at the dismount (CONTEXT_TEMPLATE.txt, SVB_PROMPT.md, svb.py), FILE_GONE and the corporate tree clean at cfd2f69 (the file was staged at 1695's BEFORE and never committed; the alias's -a cancelled the add against the rm, nothing to commit being the wanted outcome). The post for SVB-133 rides deed 1695's prompt.md and 1696's; Slack and Drive are UNWITNESSED. The reading half of the gate is the next ride's first compile: a header with no deliverable phrase and no integrity alert (1696's alert names the commented line the template still carried).

@@ -2576,6 +2577,17 @@ MATCHBOOK_CHOP = r"""

feature called a fix; e86363d "Update minirules-reference.md with
detailed deed 1643 analysis and deliverable documentation" for
two

files, both named, true.

+# SPECIMENS 2026-10-01, the door ride (pipulate, same m alias):
43006cba

+# "chore: Update test_install_doors.sh for door table testing" for
three

```

+# files (+181/-4) that CREATED the test, the door table in
install.sh
+# and the .gitignore entry unnamed; 3f09904d, a GLOSSARY.md-only
+38/-3,
+# got a two-sentence subject whose list ends on "the purpose of the
+# dotfile-blind probe", the entry under the insertion read as the
change
+# (the 0b472c12 shape); 46118ec0 "chore: Remove unused test
+# script" for a foo_files.py-only +6/-1 that CLAIMED the new test
and
+# added five TODOs, the sign read backwards; 34bbaa4b, the 2.74
bump,
+# true; 30dcbaec "chore: Update journal index with new references",
true
+# for one file of two, pyproject.toml's 2.73 to 2.74 unnamed.
# - TODO (2026-09-27, THE NAME IS THE KEY; what stayed hardwired
after -t grim landed): (a) base_url and preview_port must agree for a
blog served on nixos.local and live as two lines, so the 2026-09-27
renumber moved one and not the other until a car caught it (the
Grimoire's canonical_url would have pointed at the Vault); either
derive base_url from preview_port for local lanes (one function over
the attrset in blogs.nix) or make materializeBlogsJson refuse a local
base_url whose port disagrees with preview_port. Gate: a deliberate
mismatch makes nr refuse, or the json agrees without a hand edit. (b)
generate_ai_context.py reads -t through its own parser and
lsa.load_targets, numbers only; release runs it with -t 1, one line
when it rides. (c) The Mac shadow json numbers its lanes 1, 3, 4 with
no vault; with names the numbering stops mattering, and `vault` there
refuses with the roster until flake.nix's Darwin template grows a
fourth lane and a fourth shadow folder. (d) ~/repos/Pipulate.com/.port
still reads 4002 from before that blog left the matrix; nothing tends
it, harmless until jes runs there. (e) blogs.nix's
materializePortFiles is why the three hand-typed .port echoes of
2026-09-27 were a no-op: the switch writes them, only when they
differ; nothing to build, one thing to stop doing.
# - TODO (2026-09-26, THE EMPTY REPLACE ATE THE NEXT CAR; convicted
at deed 1596): apply.py's block_pattern closes with an optional body
group that is greedy-optional, so a deletion block written as adjacent
DIVIDER and REPLACE markers parses as empty only when it is the LAST
block in its payload; with a block behind it, the lazy body expands to
that block's REPLACE marker and the deletion's replacement becomes the
next car's whole text, markers included. The residual-marker airlock
refused the write (publishizer.py at lines 29, 32 and 35) and the file
was untouched, so the 2026-09-26 amendment held by the airlock and not
by the grammar. CURE, one character: make the group lazy-optional, a
second question mark after its closing parenthesis, so the adjacent
marker is tried first and a non-empty body still backtracks into the
group; witness it on a two-block throwaway payload whose first block

```

is a deletion, BEFORE the refusal and AFTER two APPLIED lines, then re-emit the publishizer.py cut Car 2 of the shard ride still owes (def prune_orphan_shards through its "No orphan shards." line, and the call at 172), with the deletion block LAST in its fence until the fix lands. Gate: rg -n orphan_shards scripts/articles/publishizer.py prints nothing. CURE EMITTED 2026-10-01 (deed 1750's train, Car 6: the optional group is ??, so the empty replacement is tried first); the deletion_first probe reads rc=1 file=a|b| before and rc=0 file=|B| after; the publishizer.py cut stays owed.

- TODO (2026-09-26, the renumber pass; the census that named it, deed 1595: key 3 {'Starts Late': 9, 'Gap': 9}, key 5 {'Gap': 4, 'Starts Late': 2}, no Duplicate on either side): moving 53 posts out of the Grimoire left nine days there opening late or holed and six days in the Vault the same, and the first post to arrive in the Vault from a second source makes the first Duplicate. The gauge is posts -t N -g (lsa.py, wired 2026-09-26); the actuator is a file of its own, never a --fix on the gauge: per blog and per day, order by sort_order, then git-add epoch with untracked last (incumbent wins; mv preserves mtime, so mtime cannot say who arrived), then filename; write 1..n into frontmatter only where the value changes, one line touched and mtime preserved (common.stamp_frontmatter_value is the writer); print date, slug, old -> new per rewrite; dry-run by default; agree with articleizer.py, which assigns max+1 for the day at birth; declared per blog in the blogs.nix pipeline so preview runs it. Gate: both dicts read {} after one run, and a Duplicate manufactured by a move reads {} after the next.

(nix) pipulate \$ m

 Committing: chore: Implement tests for door table testing in install.sh

[main 5dce5740] chore: Implement tests for door table testing in install.sh

1 file changed, 17 insertions(+), 5 deletions(-)

(nix) pipulate \$ git push

Enumerating objects: 24, done.

Counting objects: 100% (24/24), done.

Delta compression using up to 48 threads

Compressing objects: 100% (17/17), done.

Writing objects: 100% (17/17), 12.18 KiB | 733.00 KiB/s, done.

Total 17 (delta 11), reused 0 (delta 0), pack-reused 0 (from 0)

remote: Resolving deltas: 100% (11/11), completed with 7 local objects.

To github.com:pipulate/pipulate.git

d5085d68..5dce5740 main -> main

(nix) pipulate \$

Ignition: often "exit", "nix develop". Instructions may vary.

On branch main

Your branch is up to date with 'origin/main'.

GIT repo clean. Take BEFORE reading, make CHANGE, record AFTER diff.

```
(nix) pipulate $ exit
```

```
exit
```

```
(sys) pipulate $ nix develop
```

Already up to date.

(Nix) 2.28.5 · Python 3.12.13 · v2.74 · ~/repos/pipulate

type one:

```
menu      print this list again (useful once it scrolls away).
```

walk take guided tour of context compiler (recommended).

talk turn text-to-speech narration on or off.

connect put mcp, jira, email, docs, etc. into your contexts.

```
context  edit the list of files an AI will read (after the walk).
```

```
prompt    save your clipboard as the question for the AI.
```

```

    compile build the list and the question into one payload for a
    chatbot.

```

about how to Q/A AI output ("what Claude said").

all expanded menu.

```
(nix) pipulate $ all
```

type one:

```
menu      print the shorter list.
```

```
walk      take guided tour of context compiler (recommended).
```

talk turn text-to-speech narration on or off.

connect put mcp, jira, email, docs, etc. into your contexts.

```
context  edit the list of files an AI will read (after the walk).
```

prompt save your clipboard as the question for the AI.

compile build the list and the question into one payload for a chatbot.

about how to Q/A AI output ("what Claude said").

```
all      print this expanded list again.
```

```
brief compile this workshop into your clipboard for an AI.
```

```

jupyter start JupyterLab and Pipulate, JupyterLab first.

```

```
pu start the Pipulate server.
```

```
reseat    fix the ASCII art checksums after you edit the art.
```

```
(nix) pipulate $ osb
```

```

      \      \      \
=====<O)
      /      /      /
      the twig moved.

```

osb


```
measured) ———\
| John von Neumann worked out, on paper, a machine that builds a copy
of          |
| itself from parts it finds (Theory of Self-Reproducing Automata,
1966).      |
| Science fiction sent such machines to the stars as probes. You typed
one         |
| line. It fetched a recipe, and the recipe built this workshop from
parts on    |
| your own computer, pinned to the same versions it pins everywhere.
Java        |
| promised that in 1995: write once, run anywhere. Eelco Dolstra's
2006 PhD    |
| thesis worked out how to deliver it. The thing it describes is Nix.
|
```

```
(Enter for the next transmission, q to stop)
— TRANSMISSION 4 OF 6: THE FENCE (a story; nothing here was
measured) —
| In Michael Crichton's Jurassic Park (1990), the park's computer
counted its |
| animals by searching for the number it expected, 238, so it could
never find |
| more. Told to search for more, it kept finding more, 292 in the end:
the |
| animals were breeding. A check that cannot fail is not a check.
Every check |
| here is built so that it can fail, and apply.py, the fence around
every |
| edit, is short enough to read in one sitting. Read it.
|
|
```

```
(Enter for the next transmission, q to stop)
— TRANSMISSION 5 OF 6: THE PRIMER    (a story; nothing here was
measured) —
| In Neal Stephenson's The Diamond Age (1995), a girl named Nell is
raised by |
| a book that talks back, the Young Lady's Illustrated Primer. It
worked for |
| her because a person, Miranda, voiced it from far away. The voice in
this |
| workshop is Piper, a small text-to-speech program, and it says that
it is |
| reading a script, because a voice should say who is talking.
|
|
```

```

    (Enter for the next transmission, q to stop)
┌— TRANSMISSION 6 OF 6: THE TORTOISE    (a story; nothing here was
measured) —┐
| In 1895 Lewis Carroll wrote a short dialogue in which a Tortoise
will not    |
| accept a conclusion until Achilles writes the rule that leads to it
down as    |
| one more premise, and then the rule for that one, forever. Douglas
|
| Hofstadter put the pair back to work in Gödel, Escher, Bach (1979).
Asking    |
| one AI to check another AI's answer is the Tortoise's game: one more
premise    |
| every time. The way out is to run something and read what it
printed. That    |
| is what a receipt is.
|

```

```

So'wI' yIchu'. (Klingon: engage the cloaking device.)
Type menu to go back to being a twig.
(nix) pipulate $

```

Well that was impressive!

4: PROMPT:

Read this compile's AFTER receipts against your last reply's forecasts: the card probe prints its four lines and rc=0; the egg probe prints its title and nine rows, door none, rc=0; the census prints six lines at 977, 1056, 1062, 1143, 1144 and 1641, rc=0; and Coverage reads 75.7% (212/280). My Mac witness is pasted at the bottom.

1. If the paste shows the qamy card, the two welcome lines and the welcomed= line, mark THE DOOR SPEAKS witnessed in foo_files.py and add the Mac receipt to THE DOOR NAMES THE ROW entry in GLOSSARY.md. If any of the three is missing, name which one and find the cause before patching anything.
2. Put osb on the menu: ("osb", "operation stick bug.") as the last row of ALL_WORDS in scripts/boot_menu.py, which rides this compile, and retire THE EGG ON THE MENU.
3. My rulings: THE SILENT YES (after the question plays, print and never speak: Heard nothing? Type n.): ____ THE BANNER AND THE LABEL (a door's name on its install banner and app label, through a display-name field in door_row): ____

Mac witness paste:

```

(nix:nix-shell-env) (nix) qamyai $ exit
exit

```

Saving session...

...saving history...truncating history files...

...completed.

michaellevin@MichaelMacBook-Pro ~ % cd ~/qamyai && nix develop

```

Restored session: Thu Oct  1 23:12:38 EDT 2026
Updates found. Pulling latest changes...
From github.com:pipulate/pipulate
* branch          main          -> FETCH_HEAD
Updating 30dcbae..5dce574
Fast-forward
 GLOSSARY.md          | 30 ++++++-----
 flake.nix            | 44 ++++++-----
 foo_files.py         | 24 ++++++---
 imports/voice_synthesis.py | 167
+++++
+----
 scripts/stick_bug.py | 234
+++++
+++++
 5 files changed, 475 insertions(+), 24 deletions(-)
 create mode 100644 scripts/stick_bug.py
Update complete!
(Determinate Nix 3.19.1) 2.34.6 · Python 3.12.13 · v2.74 · ~/qamyai

```

type one:

```

menu      print this list again (useful once it scrolls away).
walk      take guided tour of context compiler (recommended).
talk      turn text-to-speech narration on or off.
connect   put mcp, jira, email, docs, etc. into your contexts.
context   edit the list of files an AI will read (after the walk).
prompt    save your clipboard as the question for the AI.
compile   build the list and the question into one payload for a
chatbot.

about     how to Q/A AI output ("what Claude said").
all       expanded menu.
(nix:nix-shell-env) (nix) qamyai $ mv ~/.config/pipulate/voice
~/.config/pipulate/voice.was && nix develop
● You are already inside the Pipulate workshop -- no need to run 'nix
develop' again.
  • Restart the server after Ctrl+C: python server.py
  • Leave this environment entirely: exit
(nix:nix-shell-env) (nix) qamyai $ exit
exit

```

```

Saving session...
...saving history...truncating history files...
...completed.
michaellevin@MichaelMacBook-Pro qamyai % mv ~/.config/pipulate/voice
~/.config/pipulate/voice.was && nix develop
mv: /Users/michaellevin/.config/pipulate/voice: No such file or
directory
michaellevin@MichaelMacBook-Pro qamyai % cd ..

```

```
michaellevin@MichaelMacBook-Pro ~ % mv ~/.config/pipulate/voice
~/.config/pipulate/voice.was && nix develop
mv: /Users/michaellevin/.config/pipulate/voice: No such file or
directory
michaellevin@MichaelMacBook-Pro ~ % nix develop
path "/Users/michaellevin" does not contain a 'flake.nix', searching
up
error: could not find a flake.nix file
michaellevin@MichaelMacBook-Pro ~ % cd qamyai
michaellevin@MichaelMacBook-Pro qamyai % nix develop
Restored session: Thu Oct  1 23:13:52 EDT 2026
Already up to date.
(Determinate Nix 3.19.1) 2.34.6 · Python 3.12.13 · v2.74 · ~/qamyai
This LLM Quality Assurance workshop talks you through steps.
This is only text-to-speech. Local AI detected but not used.
If you can hear this, press Enter. Type n to keep it quiet.
Enter or n:
Voice on. Type talk any time to turn it off.
Welcome. Ready to Q/A your AI with the Flight Data Recorder (FDR)
Different from Cockpit Voice Recorder (CVR) or "what Claude said".
```

type one:

```
menu      print this list again (useful once it scrolls away).
walk      take guided tour of context compiler (recommended).
talk      turn text-to-speech narration on or off.
connect   put mcp, jira, email, docs, etc. into your contexts.
context   edit the list of files an AI will read (after the walk).
prompt    save your clipboard as the question for the AI.
compile   build the list and the question into one payload for a
chatbot.
about     how to Q/A AI output ("what Claude said").
all       expanded menu.
(nix:nix-shell-env) (nix) qamyai $ tail -2 .door
# Written by first_contact() the day this door's welcome was shown.
Delete the welcomed line to see it again.
welcomed=2026-10-01
(nix:nix-shell-env) (nix) qamyai $ exit
exit
```

Saving session...

```
...saving history...truncating history files...
...completed.
```

```
michaellevin@MichaelMacBook-Pro qamyai % nix develop
Restored session: Thu Oct  1 23:15:54 EDT 2026
Already up to date.
(Determinate Nix 3.19.1) 2.34.6 · Python 3.12.13 · v2.74 · ~/qamyai
```

type one:

```

menu      print this list again (useful once it scrolls away).
walk      take guided tour of context compiler (recommended).
talk      turn text-to-speech narration on or off.
connect   put mcp, jira, email, docs, etc. into your contexts.
context   edit the list of files an AI will read (after the walk).
prompt    save your clipboard as the question for the AI.
compile   build the list and the question into one payload for a
chatbot.
  about    how to Q/A AI output ("what Claude said").
  all      expanded menu.
(nix:nix-shell-env) (nix) qamyai $

```

Yep, there we go.

5: DELIVERABLE: This.

Hop off the ride. This ride's stated goal is reached -- dismount. This is the NOTARY BEAT: the ride ends here, is witnessed here, and is sealed here. Answer all seven beats, briefly:

0. **TL;DR:** a short, dry, neutral abstract for the TOP of the published article -- written for an unfamiliar reader or AI summarizer who has never seen this system. No hype, no insider handles unexplained.
1. **VERIFY:** restate the goal from the top of this article and confirm (or deny) it was met, citing THIS compile's receipts, not memory. Name any ignition this ride required that never fired -- an AFTER tap taken without one is a stale BEFORE wearing the AFTER's label.
2. **BANK:** name everything that graduates -- rule, earmark, todo, pin -- as SEARCH/REPLACE patch cars against raw source present in THIS context (Target line inside the fence, one car per commit story), deletions included. A 'paste-ready' line with no Target and no SEARCH anchor is not banked; it is a hand edit the operator will not make. These BANK cars are the ONLY patches a dismount emits.
3. **DANGLING:** what carries forward unbanked? One line each, no essays.
4. **SEED:** the context.txt lines (and TODO_SLUGS if narrative context is needed) for the next ride's first compile.
5. **CLOSING:** a closing summary for the BOTTOM of the article -- the final take-away, tied to the book's larger arc where it fits naturally, never forced. Storytelling over inventory.
6. **NOTARIZE:** the artifact of record is the cartridge THIS compile writes after this payload is assembled, so its name is not in your context and you must not guess one. Attest the FINGERPRINT instead: this Manifest's wc -c line, its receipt count, and the newest receipt's date -- pure functions of the bytes the cartridge seals. Close with one line the operator completes after the compile: Deed: qamy.ai_NNNN-hhhhhhhh.zip (ls -t qamy.ai_*.zip | head -1) The seal is the signature; the fingerprint is how the deed is found.

FINALITY: after beat 6, this discussion is CLOSED. Emit NO five-car train, NO probes, NO patches beyond the BANK cars of beat 2, and NO next-turn prompt beyond the SEED lines in beat 4. Any reader or model encountering this article later should treat it as a finished, notarized document -- an archive entry, not an open thread.

Opus 5.5 Max: Rehearsing the bank cars against a fresh pull of the public repo first, then writing the seven beats.

Context verified. The bank cars touch boot_menu.py, foo_files.py and GLOSSARY.md, and all three rode deed 1779. The public repo at 5dce5740 matches this Manifest's byte counts, so I

rehearsed the cars on a copy of it.

1. VERIFY

The goal, from the top of the article: one installer known by several names. Each door gets its own folder and first words: a workshop line, a local-AI line that claims only what a lookup found, two welcome lines and a tagline, all kept in one small table.

Met for qamy.ai. The evidence in this compile: - **Compile lane, all rc=0.** - The card probe printed the door card, the old card unchanged when there's no door, Ready to QA your AI, two, and welcomed among the .door keys. - The census found the gate (lines 1056 and 1062), the spelled folder (1143, 1144), the Lix line (977) and osb (1641). - The egg printed nine readings. - Your BEFORE run at the same prompt read rc=1, 2 and 1. - Coverage read 75.7% (212/280), as forecast, and Ruff is clean. - **The Mac paste in this Prompt.** It shows: - the qamy workshop line and Local AI detected but not used.; - the question, then Voice on.; - both welcome lines; - welcomed=2026-10-01 in .door; - a following entry with only the menu.

The folder half (Qamyai -> ~/qamyai) is GLOSSARY's deed-1778 receipt, which also rides in this compile. - **Version lines.** The Mac's entry line read (Determinate Nix 3.19.1) 2.34.6 under the old runScript (23:12) and under the new one (23:13, 23:15). Prime read (Nix) 2.28.5 under the new one. Prime's earlier reading isn't in this payload. - **No download lines.** Neither the 60 MB line nor Voice ready. printed. Both print only on a download (voice_synthesis.py lines 502 and 527), and the Mac already had the model.

Ignitions that never fired, so nothing is claimed for them: - jn on an install not yet onboarded, so the spelled folder is unheard. - A recorded answer meeting a fresh .door, the welcomed branch. - The first entry after the pull ran the runScript built before the pull. - The voice file was moved aside before the next entry, so that entry took the question path instead. - A Lix machine. - A real install through npvg.org or pipulate.com. Those rows have only passed the stub test. - A fresh machine, which is what the download lines under a door's card need. - all after Car 1. The osb row has only run in my sandbox. - Sound. Enter counts as yes whether or not anything was heard.

2. BANK

Graduating: - THE EGG ON THE MENU: the TODO is deleted and its row lands. - Stick Bug: PENDING becomes WITNESSED. - THE BLANK STARE RULE: its one exception is named. - THE DOOR SPEAKS: WITNESSED, with the remainder kept. - THE VERSION LINE UNDER LIX: two readings added; Lix is still owed. - One RECEIPTS line. - The ride's commit-subject specimens. - THE STEP THAT RUNS ONCE: a new EARMARK.

How I rehearsed them, on a copy of the public repo at 5dce5740: - All 10 blocks applied, in order and in reverse, and gave identical bytes. - Ruff is clean, and --recall --all ends on osb. - RECEIPTS counts 21 against its cap of 20. The five dismounts before this one each added a receipt line and deleted none, leaving the fade to forget, so this one does the same. - Pasting Car 1 twice prints one SEARCH miss, for its deletion. apply.py can't call a deletion "applied", and nothing changes.

Car 1: osb on the menu. Expect 4 applied. git counts +16/-8 across three files, or +17/-9 if the Coverage line this compile wrote is still uncommitted.

Target: scripts/boot_menu.py

```
[[ [SEARCH] ]]
```

```

    ("reseal", "fix the ASCII art checksums after you edit the art."),
)
[[[DIVIDER]]]
    ("reseal", "fix the ASCII art checksums after you edit the art."),
    # OPERATION STICK BUG (2026-10-02): osb is THE BLANK STARE RULE's
one
    # exception, a twig of a name over the one loud screen, so it
rides
    # this list, last, and the short list never prints it.
    ("osb", "operation stick bug."),
)
[[[REPLACE]]]
Target: foo_files.py
[[[SEARCH]]]
# - TODO (2026-10-02, THE EGG ON THE MENU): osb, operation stick bug,
is a word after ignition (flake.nix's osb function runs
scripts/stick_bug.py), and its menu line is owed: ("osb", "operation
stick bug.") as the last row of ALL_WORDS in scripts/boot_menu.py, so
all lists it and the short menu never does. boot_menu.py did not ride
deed 1778, and a car patches only what rode (checklist item 11), so
the line waits for the compile that carries it.
# - TODO (2026-10-02, ONE GRAMMAR FOR THE SLASH, unasked):
[[[DIVIDER]]]
# - TODO (2026-10-02, ONE GRAMMAR FOR THE SLASH, unasked):
[[[REPLACE]]]
Target: GLOSSARY.md
[[[SEARCH]]]
    AMENDED 2026-10-02 (PENDING its menu line and a witness): osb is its
word, the one abbreviation the workshop prints on purpose, THE BLANK
STARE RULE's single exception, because looking like a twig is the
camouflage. Typed, it runs scripts/stick_bug.py, the one loud screen
in
    a quiet workshop: readings taken at the moment they print, then six
transmissions that say they are a story.
[[[DIVIDER]]]
    AMENDED 2026-10-02 (WITNESSED at deed 1779): osb is its word, the
one
    abbreviation the workshop prints on purpose, THE BLANK STARE RULE's
single exception, because looking like a twig is the camouflage.
Typed,
    it runs scripts/stick_bug.py, the one loud screen in a quiet
workshop:
    readings taken at the moment they print, then six transmissions that
say they are a story. THE WITNESS: typed by hand on Prime, it
printed
    the moving twig, the banner, nine readings (clock 1,790,910,458) and
all six transmissions, ending on Type menu to go back to being a
twig.; in the compile lane, scripts/stick_bug.py --readings printed

```

the
 same nine rows ten minutes later, rc=0. Its menu line, the last row
 of
 all, rode the dismount's car; the next all is its witness.
 [[[REPLACE]]]
 Target: GLOSSARY.md
 [[[SEARCH]]]
 Operation Stick Bug's rule for names: the instrument vanishes when it
 is called what it is. Sibling of THE NEXT ACTION, NOT THE MACHINERY.
 [[[DIVIDER]]]
 Operation Stick Bug's rule for names: the instrument vanishes when it
 is called what it is. Sibling of THE NEXT ACTION, NOT THE MACHINERY.
 ONE EXCEPTION, on purpose (2026-10-02): osb, Operation Stick Bug's own
 word, printed last by all and never by the short menu, because a name
 like a twig is its camouflage.
 [[[REPLACE]]]

Car 2: THE DOOR SPEAKS, witnessed. Expect 3 applied. git counts +13/-3 across two files.

Target: GLOSSARY.md
 [[[SEARCH]]]
 along unread at 2.74, so the old card played; THE DOOR SPEAKS is
 their
 reader.
 [[[DIVIDER]]]
 along unread at 2.74, so the old card played; THE DOOR SPEAKS is
 their
 reader. THE WORDS, WITNESSED (deed 1779, the same Mac): the first
 entry
 after the pull printed only the menu although .door had no welcomed=
 line, which the new gate opens for, so the runScript that ran was
 the
 one built before the pull. With the voice answer moved aside, the
 next
 entry printed This LLM Quality Assurance workshop talks you through
 steps., This is only text-to-speech. Local AI detected but not
 used.,
 the question and Enter or n:; Enter printed Voice on. and both qamy
 welcome lines; tail -2 .door read the comment and
 welcomed=2026-10-01,
 the Mac's local date at 23:13 EDT, a day behind UTC; the entry after
 that printed only the menu. A paste carries no sound, and Enter is
 yes
 whether or not anything was heard (THE SILENT YES).
 [[[REPLACE]]]
 Target: foo_files.py
 [[[SEARCH]]]
 # - TODO (2026-10-02, THE DOOR SPEAKS, LANDED by the reply to deed
 1778, WITNESS OWED): first_contact() in imports/voice_synthesis.py

reads .door through read_door() and says that door's card (door_card: the workshop line, then This is only text-to-speech. with local_ai_line()'s No local AI found yet. or Local AI detected but not used., then the question); after the answer it prints the welcome lines and speaks them only on yes (_show_welcome; _spoken says Q/A as QA, read through the phonemizer inside piper-tts 1.6.0 in a sandbox), then writes welcomed= into .door. flake.nix's gate also opens for a .door with no welcomed= line, so a second install on a computer that already answered shows its workshop line and welcome once, unasked. With no .door the card is unchanged, byte for byte, in a sandbox run against a fake player. All five rulings landed; ruling 5 is TTS_DIR_SPELLER in flake.nix. WITNESS OWED, on the Mac: ~/qamyai's second entry after the push (the first one pulls), with ~/.config/pipulate/voice moved aside, shows and says the qamy card and welcome, and a third entry shows neither. The spelled folder is heard on the first jn of an install not yet onboarded. THE SILENT YES stands under the new words.

[[[DIVIDER]]]

- TODO (2026-10-02, THE DOOR SPEAKS, WITNESSED at deed 1779 on the Mac; the receipt is THE WORDS, WITNESSED in GLOSSARY.md's The Door Names the Folder): first_contact() in imports/voice_synthesis.py reads .door (read_door), says the door's card (door_card, local_ai_line), prints the welcome and speaks it only on yes (_show_welcome, _spoken), and writes welcomed= into .door (_mark_door_welcomed); flake.nix's gate also opens for a .door with no welcomed= line. All five rulings landed. STILL OWED, each by hand: the welcomed branch (a computer that already answered meets a fresh door: no question, the workshop line and the welcome once), n and silent with a door, the spelled folder heard (TTS_DIR_SPELLER, the first jn of an install not yet onboarded), the download lines under a door's workshop line (the Mac already held the model), npvg.org's and pipulate.com's cards at a real door, and any sound at all, which no paste carries; THE SILENT YES stands under the new words.

[[[REPLACE]]]

Target: foo_files.py

[[[SEARCH]]]

Nix and Determinate Nix print what they printed before, and the witness is the entry line on the first Lix machine.

[[[DIVIDER]]]

Nix and Determinate Nix print what they printed before (READ at deed 1779: the Mac's entry line read (Determinate Nix 3.19.1) 2.34.6 under the old line and the new, and Prime's read (Nix) 2.28.5 under the new), and the witness is the entry line on the first Lix machine.

[[[REPLACE]]]

Car 3: the ride's ledger. Expect 3 applied. git counts +16/-0.

Target: foo_files.py

[[[SEARCH]]]

```

# --- START RECEIPTS (newest first; cap 20 lines; a line pushed past
the cap is deleted, never moved -- git and the rolling pin are the
archive) ---
[[[DIVIDER]]]
# --- START RECEIPTS (newest first; cap 20 lines; a line pushed past
the cap is deleted, never moved -- git and the rolling pin are the
archive) ---
# 2026-10-02 dismount THE DOOR SPEAKS (deeds 1773, 1778 and 1779, 1774
through 1777 compiles that rode no turn, INFERRED; pipulate commits
43006cba the door table, 3f09904d the glossary, 46118ec0 the router,
34bbaa4b and 30dcbaec the 2.74 Doorways release, cut ahead of the
router's HOLD THE TRUCK note, which gave the folder half its witness,
d5085d68 the boundary, 868425bd the reader, 869637dc the gate, the
spelled folder, the Lix line and osb, daaac215 stick_bug.py, 8149a74c
the glossary, 5dce5740 the router, the five cars pushed by hand): one
installer, three doors; the stamp picks a row in install.sh, the row
names the folder (qamy.ai installs ~/qamyai) and writes the door's
words to .door, and first_contact() says them. WITNESSED: Qamyai ->
~/qamyai at 1778; at 1779 the Mac's door card, both welcome lines,
welcomed=2026-10-01 and a menu-only entry after it, the first entry
after the pull having run the runScript built before it; the Mac's
entry line the same under the old version sed and the new, Prime's
(Nix) 2.28.5 under the new; osb by hand on Prime and its readings in
the compile lane; BEFORE rc=1, 2 and 1, AFTER 0, 0 and 0. MISSES:
Topological Integrity forecast 8, read 15 (it counts commented paths);
welcomed= forecast from the UTC date, a day ahead of the Mac's; Step 2
chained a one-time mv to nix develop with &&, so one paste at the
wrong prompt cost four tries (THE STEP THAT RUNS ONCE). UNWITNESSED:
the welcomed branch, n and silent with a door, the download lines at a
door, the spelled folder heard, any sound, Lix, and the npvg.org and
pipulate.com cards at a real door. The block now reads 21 against its
cap of 20; the next forget fades the bottom line.
[[[REPLACE]]]
Target: foo_files.py
[[[SEARCH]]]
Ruling owed: should a door's app carry the door's name.
[[[DIVIDER]]]
Ruling owed: should a door's app carry the door's name.
# - EARMARK: THE STEP THAT RUNS ONCE (banked 2026-10-02,
self-convicted at deed 1779): a witness step never chains a one-time
change to its reading with &&. Step 2 read mv ~/.config/pipulate/voice
~/.config/pipulate/voice.was && nix develop; pasted inside the nix
shell, the mv ran and nix develop was refused, and every paste after
exit stopped at mv, so the witness took four tries and left voice.was
on the Mac. The change goes on its own line, written so a second paste
does no harm, and the reading on the next.
[[[REPLACE]]]
Target: foo_files.py

```

```

[[[SEARCH]]]
#   true; 30dcbac "chore: Update journal index with new references",
true
#   for one file of two, pyproject.toml's 2.73 to 2.74 unnamed.
[[[DIVIDER]]]
#   true; 30dcbac "chore: Update journal index with new references",
true
#   for one file of two, pyproject.toml's 2.73 to 2.74 unnamed.
#   SPECIMENS 2026-10-01, the door ride's second train (pipulate, same
m
#   alias): d5085d68, the boundary before the cars, foo_files.py-only
and
#   +1/-1 (the Coverage counter), got the subject flake.nix's router
hint
#   prescribes, word for word, true; 868425bd "feat: add .door read
and
#   first_contact() implementation" true; 869637dc "chore: Update Nix
#   version reading and voice welcome message" roughly true, osb
unnamed;
#   daaac215 "chore: Update stick_bug.py with more detailed
explanations
#   and improvements" for a file CREATED (+234/-0); 8149a74c named
"Stick
#   Bug" truly and "The Dotfile-Blind Probe", the entry under the door
#   insertion, as the change, as 3f09904d had earlier the same evening
(the
#   0b472c12 shape twice in one ride); 5dce5740, foo_files.py-only
like
#   d5085d68 but carrying TODO text, got "chore: Implement tests for
door
#   table testing in install.sh", the specimen it added quoting
#   43006cba's subject read as the change (the 6f09e025 shape).
[[[REPLACE]]]

```

3. DANGLING

- THE SILENT YES: ruling owed; you left it blank at 1779.
- THE BANNER AND THE LABEL: ruling owed; you left it blank at 1779.
- The welcomed, n and silent branches with a door: not yet seen on a real machine.
- The spelled folder: unheard until the first jn of an install not yet onboarded.
- The download lines under a door's card: not yet seen, because the Mac already had the model.
- Lix: no machine to test on yet.
- npvg.org and pipulate.com: their rows pass the stub test only.
- ~/.config/pipulate/voice.was is still on the Mac, left over from Step 2. The live answer is ~/.config/pipulate/voice. To remove the leftover, at michaellevin@MichaelMacBook-Pro ~

- `% run rm ~/.config/pipulate/voice.was.`
- RECEIPTS reads 21 against its cap of 20 after Car 3. The next forget fades THE ACTUATOR IS NOT THE LEDGER.
- ONE GRAMMAR FOR THE SLASH: waits until a second spoken line has a slash.
- PyPI 2.74 predates the voice half. Door installs pull main, so the next release is your call.
- NOTARY FINGERPRINT: the template asks for a `wc -c` line the compiler doesn't emit, and `init.lua`'s "must not guess" line is out of date now that the seal footer names the deed. Both are already router TODOs.

4. SEED

```
! cd "$PIPULATE_ROOT" && pwd
! python scripts/boot_menu.py --recall --all; echo "rc=$?"
! bash tests/test_install_doors.sh; echo "rc=$?"
! awk '/^# --- START RECEIPTS/{f=1;next} /^# --- END RECEIPTS/{f=0} f'
foo_files.py | wc -l
imports/voice_synthesis.py
assets/installer/install.sh
flake.nix
scripts/boot_menu.py
foo_files.py
GLOSSARY.md
```

No TODO_SLUGS: the open TODOs ride in `foo_files.py`.

5. CLOSING

The ride started with a name that sounded wrong out loud. Read by a voice, `~/qamy` comes out as "commie" and `~/qamyai` comes out as "QA my AI". Fixing it meant pulling apart three things that had been one string: the address a stranger types, the folder the installer makes, and the first sentences the workshop says. They now sit in one small table, and the web server picks the row by changing a single word on the way out.

None of it was accepted because a model said it worked, including this one. Each claim went through the same loop: read the machine, change one thing, read it again, and keep both printouts. That loop is the answer to Lewis Carroll's Tortoise, who will accept no conclusion without one more premise: stop adding premises and run something. Where nothing printed (the sound, the other two doors, a Lix machine), the record says "not yet". `osb`, the one loud screen, follows the same rule: it takes its readings as it prints them, and its stories say they are stories. Done every turn, the loop stops needing thought, and the article that opened this ride has a word for that: myelination.

6. NOTARIZE

- **Fingerprint:** no `wc -c` line rides this Manifest, because the compiler doesn't emit one. It does emit Total Bytes 1,329,846 (UTF-8) and Verified Tokens 336,904.
- **Receipts:** 4 LIVE COMMAND RECEIPTS.
- **Newest dated reading:** the egg probe's clock, 1,790,911,062, which is

2026-10-02T03:17:42Z (23:17:42 EDT on 10-01). It is the only date any receipt prints.

- **Deed:** read off the compiler's seal footer, which sits outside the sealed bytes, so it isn't a guess.

Deed: qamy.ai_1779-a17b0730.zip (ls -t qamy.ai_*.zip | head -1)

Book Analysis

Ai Editorial Take

What is most surprising about this piece is how a purely phonetic problem—text-to-speech mispronouncing a folder path as 'commie'—acted as the catalyst for an entire architectural decoupling of domain names from filesystem structures. It illustrates that when you introduce real multi-modal feedback (local voice synthesis via Piper) into a developer workflow, the sensory affordances of speech enforce higher standards of code hygiene than silent text ever did. The software had to become cleaner simply so it would sound honest when spoken aloud.

Content Potential And Polish

- **Core Strengths:**
- Brilliant cognitive framing of myelination (literate, automatic, prosodic, fluent) that explains why SaaS churn keeps developers perpetually deskilled.
- Compelling use of the Flight Data Recorder (FDR) versus Cockpit Voice Recorder (CVR) distinction to explain the difference between verifiable runtime telemetry and LLM conversational outputs.
- Exceptional use of literary and intellectual archetypes (Hofstadter, Carroll, Crichton, Stephenson) to illuminate modern software engineering dilemmas.
- Rigorous empirical demonstration: actual test scripts, before-and-after diffs, and cross-platform verification between Linux and macOS.
- **Suggestions For Polish:**
- Provide a quick visual table or callout early in the text defining FDR vs. CVR to orient newcomers before they encounter the technical git diffs.
- Smooth the transition into the Achilles and the Tortoise dialogue so casual readers understand that the model has taken on the Tortoise persona to pressure-test the author's claims.
- Explicitly mention how the .door dotfile isolation prevents config pollution across independent installations on a single workstation.

Next Step Prompts

- Draft a lightweight test matrix in bash that verifies offline fallback behavior when Piper TTS or local audio hardware is unavailable during first-contact onboarding.
- Create an end-user guide comparing Cockpit Voice Recorders (CVR) with Flight Data Recorders (FDR) to help enterprise AI teams establish verifiable audit trails for LLM code generators.