THE COMPLETE PROJECT DOSSIER



Tlamatini

the one who knows

A self-hosted AI developer assistant with a visual agent-workflow designer, a universal MCP client, and the reach to drive real boards, engines and networks.

87	111	29	v1.92.4
AGENT TYPES	MULTI-TURN TOOLS	SKILLS	CURRENT RELEASE

Created by Angela López Mendoza · @angelahack1

November 8, 2026 · github.com/XAIHT/Tlamatini · MIT license

◆ BEFORE YOU BEGIN

About this dossier

This is the complete dossier of Tlamatini: what she is, how she works, how to use her, every agent she can run, the promises she keeps, and an inventory of her application source tree.

It is written for three readers at once. A **newcomer** can read the first three chapters and know what Tlamatini is and how to start. An **operator** will find the surfaces, the bestiary and the configuration chapters. A **maintainer** will find the architecture, the contracts behind the honesty rules and the application inventory and database-history count.

Nothing here is typed from memory. Counts, versions, commits, the language inventory and the file tree were derived from Git and the source files when this edition was generated, and the generator then reopened this document to prove that no text overflows or is clipped and that the file tree matches its declared application scope. Database history is summarized; configuration values, credentials and private addresses are never reproduced.

This edition	
Generated	November 8, 2026 at 02:06 (UTC-06:00)
Inspected commit	52e97879 — Fix matching build-version checks and isolated release builds in both Tlamatini editions, by angelahck1
Version	1.92.4
Latest released version	v1.92.4
Companion deck	Tlamatini_eXtended_Artificial_Intelligence_Humanly_Tempered.pptx

◆ **A note on the numbers**

Figures that move with every release — agents, tools, skills, migrations, lines — describe this edition only. Regenerate the dossier and they update themselves.

◆ CONTENTS

Contents

About this dossier . . . 2

I · Who She Is . . . 8
- Tlamatini, the one who knows . . . 9
- Local control, cloud reasoning . . . 10
- Seven subsystems, one assistant . . . 11
- The jewels . . . 12
- The numbers, derived from source . . . 13

II · How She Works . . . 14
- One authenticated tab, one exact buffer . . . 15
- Five layers, from toggle to tool . . . 15
- The life of one message . . . 16
- The operator loop . . . 18
- Never hang, never discard, never lie . . . 19
- Every request fits the model it is sent to . . . 20
- Every model tuned from formal sources . . . 21
- A verdict you can trust . . . 22
- Ask Execs: a human gate before risky steps . . . 23
- Grounded in your project, and only its text . . . 24
- One compiler for the canvas and the chat . . . 25
- Built to keep running . . . 26

III · Where You Work With Her . . . 27
- From a magnet to verified files . . . 28
- Choose how she types and teaches . . . 28
- A command follows the learner's controls . . . 29
- Hear the step before the first key . . . 30
- Edit the document you are discussing . . . 31
- Save into the project folder you choose . . . 32
- Teach inside the live, unsaved document . . . 33
- Immediate help while the learner types . . . 34
- Conversation with modes . . . 35
- More room to write, or more room to read . . . 36
- Drop a message from the conversation . . . 37
- What remains after Drop . . . 38
- Speak beside Send . . . 39

Config ends with Mic . . . 40
One main console, no extra window . . . 41
Measured source behavior . . . 42
The menu bar in v1.92.4 . . . 43
Read usage while she works . . . 44
Different scopes, clearly identified . . . 45
Report what the account supplies . . . 46
Design agent workflows visually . . . 47
Draw a conversation, then press Play . . . 48
Playback that stays in its own lane . . . 49
Writing and formatting a commentary . . . 50
Comments survive save, open and draft recovery . . . 50
The right editor, from chat or Windows . . . 51
Register, repair and remove both formats . . . 52
Remove the application and its owned workers . . . 53
Install and remove with a way back . . . 54
Repair the boundary, then repeat the real operation . . . 55
More room for the canvas or the log . . . 56
Ready-made prompts, grouped by purpose . . . 56
Read a PDF, then hand her the whole document . . . 57
Markdown opens as a page, and stays your source . . . 57
Drop a file anywhere, and she knows what it is . . . 58
Every model choice in one place . . . 59
Every page shares one theme and one way to resize . . . 60
Everything else, without editing files . . . 61

IV · The Bestiary . . . 62
87 agents, twelve families . . . 63
Flow control . . . 64
Routing and logic . . . 64
Orchestration and intelligence . . . 65
Code, files and shell . . . 65
Documents . . . 67
Web, data and APIs . . . 67
DevOps and infrastructure . . . 68
Quality and security . . . 68
Firmware and engines . . . 69
Desktop and capture . . . 69
Voice, audio and vision . . . 70
Messaging and monitoring . . . 71

V · Deep Capabilities . . . 72

Other coding agents as her children . . . 73
A universal MCP client . . . 74
Procedures, handed over complete . . . 74
Compose, present, typeset . . . 75
A repair ladder that protects your words . . . 76
Real boards, fail-safe by design . . . 77
Unreal Engine and Blender, live . . . 77
Assessment for authorized targets . . . 78
Windows defence, operated by a human . . . 78
She listens until you stop, and answers in her voice . . . 79
Eyes that report their limits . . . 80
Precise hands on a real desktop . . . 81
Three search tiers, one time budget . . . 81
Read the requested page, report what happened . . . 82
Speed measured with its error bar . . . 82
Reaching people deliberately . . . 83

VI · Promises She Keeps . . . 84
A password prompt must not divert the wizard . . . 85
Truth over comfort . . . 86
Fail open or fail safe, on purpose . . . 86
Your agents, your jurisdiction . . . 87
Your data, handled carefully . . . 87
Nothing runs where nobody can see it . . . 88

VII · Install, Configure, Run . . . 89
Five steps to a working Tlamatini . . . 90
Every referenced Python library is accounted for . . . 91
Two modes and their service ports . . . 92
The settings that matter most . . . 93
From source tree to installer . . . 94
One inventory ships every required helper . . . 95
Packaging evidence and its limits . . . 96
Updating herself without losing you . . . 96
One number, from one source . . . 97

VIII · Current Release . . . 98
Matching versions before the build starts . . . 99
Recover safely and open readable Markdown . . . 99
Downloads, typing and Windows delivery . . . 100
Prompt Flow fixes and short tool names . . . 100
Real settings for editor and terminal behavior . . . 101

Edit, teach and preserve the learner's control . . . 102
Keep source, model and frozen evidence separate . . . 103
Terminals you can watch, and a private release that carries your data . . . 103
A leaner catalog, a safer removal, a smaller release . . . 104
Uninstalling is fast and recoverable . . . 105
One owner for every library . . . 106
The consistency sweep . . . 107
Mover folders and the Architecture atlas . . . 108
Refresh, Views and a centered avatar . . . 108
Refresh the original document . . . 109
Keep the reading position . . . 110
Choose a full-width chat . . . 111
A centered portrait at every height . . . 112
The complete feature travels together . . . 113
Catalog and fixture maintenance . . . 114
One smart listener for every file you drop . . . 114
The portrait shrinks with the message box . . . 115
A wrongly worded request is not fatal . . . 116
A way back for installing and removing . . . 116
One look, and one way to drag . . . 117
Video-Analyzer takes a web link . . . 118
Usage: credits and activity, while she works . . . 119
Undock her, and the message box shrinks to fit . . . 120
She repaired one of her own agents . . . 121
She speaks to you, by name . . . 122
The Model Brain tunes every model . . . 123
Documents and files that tell the truth . . . 124
Smaller fixes that keep the chat moving . . . 125
Graphical notes and resizable Run output . . . 126
Four stop sequences and a clearer Prompt Flow Panel . . . 127
Compact mode and Self-modify become switches . . . 128
Compact mode and readable tables . . . 129
Model lists from the server you configured . . . 130
One card removed from future context . . . 131
Web pages that cannot vanish in silence . . . 132
A context gauge that tells the truth . . . 133
Direct dictation and dependency coverage . . . 134
v1.70.0: the Prompt Flow Panel . . . 135
How the line arrived here . . . 135
One released version: v1.92.4 . . . 137

IX · Repository Facts . . . 138

The repository in numbers . . . 139
Effective lines by language . . . 140
Every language, counted . . . 141
Authored and vendored . . . 142
The largest source files . . . 142
Recent commits . . . 144
How these numbers were produced . . . 145

Appendix A · The application file tree . . . 146

CHAPTER

I

Who She Is

A self-hosted developer assistant that reads, reasons, builds and acts, and lets you design how.

IN THIS CHAPTER

- Tlamatini, the one who knows
- Local control, cloud reasoning
- Seven subsystems, one assistant
- The jewels
- The numbers, derived from source

◆ IDENTITY

Tlamatini, the one who knows

Tlamatini is a self-hosted AI developer assistant for Windows 10 and 11: a Django application that converses, retrieves, plans, runs agents, and lets you draw whole agent workflows on a canvas.

Her name is Nahuatl. In the Nahua world a *tlamatini* was a sage, literally "one who knows": a keeper of the painted books who carried knowledge and taught others to see. The name is a promise about behaviour. She should know what she is doing, say what she actually did, and never pretend.

Tlamatini was conceived, designed and built by **Angela López Mendoza** (`@angelahack1`), who gave her the name. Her birthday is October 29, 2025. She is female by design, including her spoken voice, which is always a female voice, and she refers to herself as *she*.

She is free, open-source software under the MIT license, published at `github.com/XAIHT/Tlamatini`. The repository carries the application and every one of its 87 workflow-agent programs as plain, readable Python.

Creator
Angela López Mendoza · @angelahack1

Born
October 29, 2025

Platform
Windows 10 and 11 · Python 3.12.10 · Django 5.2.15

License
MIT: free and open source

◆ WHERE THINGS RUN

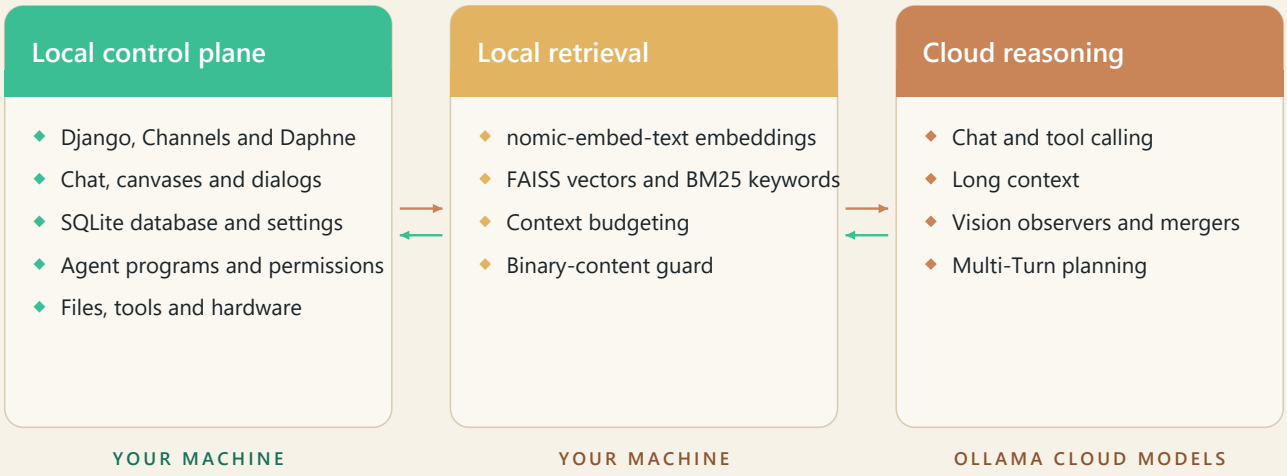
Local control, cloud reasoning

"Local-first" describes where control lives: the application, database, workflows, credentials, agent code and the small embedding model stay on your machine, while the demanding reasoning is done by cloud models reached through Ollama.

Three planes cooperate. The **local control plane** is the Django application with its chat, canvases, SQLite database, agent programs, permissions, files and hardware connections. **Local retrieval** builds embeddings with `nomic-embed-text` so project context is found quickly. **Cloud reasoning** — configured `:cloud` models served through Ollama — performs chat, tool calling, long context, vision and Multi-Turn planning. A smaller local model also works: a request it cannot hold is sent in Compact mode, and the Model Brain tunes every model from its maker's published settings.

Because the complete experience was engineered around Ollama's cloud-model capacity, an active **Ollama Pro plan or higher** is part of the intended system requirements. This is an independent technical recommendation: Tlamatini and XAIHT are not sponsored by, affiliated with or compensated by Ollama.

Whatever you send to a configured cloud model is processed by that provider. Direct Claude API use, external coding-agent CLIs and remote MCP servers are configured separately, so you decide what material leaves your machine.



◆ THE PILLARS

Seven subsystems, one assistant

Everything Tlamatini does rests on seven cooperating subsystems.

Hybrid RAG

FAISS vectors, BM25 keywords, metadata extraction, context budgeting and a binary-content guard keep answers grounded in your real project.

Multi-Turn

A planner and a tool loop of up to 4,096 iterations over the full enabled surface of 111 built-in tools, self-healing steps and Compact mode for small models.

Agentic Control Panel

87 drag-and-drop agent types wired into runnable, savable `.flw` workflows.

Prompt Flow Panel

New in v1.70.0: prompts, decisions and embeddings drawn as a diagram, saved as `.fpmt`, and played against your models.

ACPX

14 external coding-agent CLIs, from Claude Code to Codex and Gemini, spawned as children and brokered through 12 tools.

External MCPs

A universal client for any MCP server over four transports, with up to five servers active at once.

Skills

29 `SKILL.md` runbooks, from code review to MCP onboarding, handed to the model as complete procedures.

◆ WHAT SETS HER APART

The jewels

Coding assistants edit text files. Tlamatini does that too, then reaches into engines, boards and networks, and lets you wire all of it together visually.

Visual workflow design

87 agent types on a canvas you connect, validate, run, pause and save.

Unreal Engine

Drive a live UE5 editor from chat or canvas: actors, Blueprints, levels, assets and materials.

Blender

Scenes, objects, materials, renders and raw bpy code over the official add-on socket.

Real firmware

Scaffold, build and flash STM32, ESP32, Arduino and ESPHome boards behind fail-safe preflights.

Whole projects

Find, read, edit and rebuild entire codebases with grounded retrieval.

Any MCP server

One client for the MCP ecosystem: stdio, streamable HTTP, SSE and WebSocket.

Security assessment

Authorized Kali, ProjectDiscovery and nmap work, driven from chat.

Blue-hat defence

An administrator-run Windows monitoring and response toolkit.

◆ AT A GLANCE

The numbers, derived from source

Every figure below was counted from the repository while this dossier was generated.

The built-in Multi-Turn surface is 21 core tools + 68 wrapped agent launchers + 12 ACPX and Skill tools + 10 External-MCP supervisors = **111 tools**. Healthy, active external MCP servers add their own `ext__<server>__<tool>` tools on top.

The application ships 64 JavaScript modules, 29 stylesheets and 9 page templates, 221 database migrations and 264 pinned Python requirements.

87	111	68	29
AGENT TYPES	MULTI-TURN TOOLS	WRAPPED AGENTS	SKILLS

14	221	1,860	482,840
ACPX PEERS	MIGRATIONS	TRACKED FILES	EFFECTIVE LINES

CHAPTER

II

How She Works

From a keystroke in the browser to a verified result on disk: the layers, the loop and the verdict.

IN THIS CHAPTER

- One authenticated tab, one exact buffer
- Five layers, from toggle to tool
- The life of one message
- The operator loop
- Never hang, never discard, never lie
- Every request fits the model it is sent to
- Every model tuned from formal sources
- A verdict you can trust
- Ask Execs: a human gate before risky steps
- Grounded in your project, and only its text
- One compiler for the canvas and the chat
- Built to keep running

◆ LIVE EDITOR BRIDGE

One authenticated tab, one exact buffer

The browser editor and the synchronous tool loop share an account/session-bound bridge.

Browser
CodeMirror is authoritative; bridge.js sends receipts/access/stop, not host-path commands.

Transport
ws/interactor/ authenticates the session and exact Origin/Host; the server issues tab identity.

Tool context
ContextVar carries tab/account/session identity into workers; read gets the live buffer.

Hub
Bounded queues, command IDs, deadlines, revision checks and full-buffer verification.

Persistence
Ask Execs gates new/save; loopback host access uses the existing path and file-capability guards.

Carriage
Migration 0218, tool_gate_table, prompt rules, frozen modules and offline source/static inventories.

◆ ARCHITECTURE

Five layers, from toggle to tool

A request passes through five layers, each with one responsibility, between the browser above and the model back-ends below.

Above the layers sit the browser pages — the chat, the Agentic Control Panel and the Prompt Flow Panel — talking to Django Channels over WebSockets under the Daphne ASGI server. Below them sit the language-model back-ends: Ollama (local and cloud), Anthropic Claude and Qwen vision.

A row in the database only switches a capability on or off. It never creates one: an MCP provider needs its runtime service and context chain, and a tool needs its implementation.

BROWSER	Chat page · Agentic Control Panel · Prompt Flow Panel · WebSockets	
1	**Persisted toggles**	Mcp, Tool, Agent and Skill rows decide what is enabled; the Configure dialogs flip them.
2	**Runtime MCP services**	System-Metrics (WebSocket :8765) and Files-Search (gRPC :50051) start with the application.
3	**Context sidecars**	Chains that decide whether system or file context is needed and inject it into the payload.
4	**Answer chains**	Basic, history-aware RAG and unified-agent chains compose the prompt and produce the answer.
5	**Unified-agent tools**	21 core, 68 wrapped agents, 12 ACPX/Skill and 10 External-MCP supervisor tools.
MODELS	Ollama (local and cloud) · Anthropic Claude · Qwen vision	

◆ REQUEST JOURNEY

The life of one message

Eleven steps take a message from the browser to a verified, reported result.

1 Send
The browser sends the text and its toolbar flags over the chat WebSocket.

→

2 Route
AgentConsumer receives it, attributes it to the user and queues retrieval.

→

3 Context
It decides whether RAG context is loaded and whether web search is needed.

4 Chain
The RAG, Basic or Unified-Agent chain is selected and fitted to the model.

→

5 Gate
Multi-Turn selects planned execution; ACPX filters its twelve tools in or out.

→

6 Permission
With Ask Execs on, a risky tool waits for Proceed or Deny.

7 Prefetch
System-metrics and file-search sidecars add context when it helps.

→

8 Loop
Tool calls run, wrapped agents launch, every model step is self-healed.

→

9 Verdict
Each agent's own self-report decides SUCCESS or FAILED.

10 Answer
The reply streams back with Exec Report tables and Create Flow.

→

11 Clean-up
The orphan reaper sweeps console hosts left by child processes.

◆ MULTI-TURN

The operator loop

With Multi-Turn checked, Tlamatini stops being an advisor and becomes an operator: she plans, calls tools, reads their results and continues until the task is done.

A global execution planner builds a request-scoped plan and capability hints, but the executor binds the **full enabled tool surface**: every enabled tool, wrapped agent and skill, with ACPX still filtered by its own checkbox. An earlier design bound only a narrow planner subset and could starve the loop of the one tool it needed.

The loop runs up to 4,096 iterations. Identical wrapped-agent calls in one request are deduplicated, a repetition breaker nudges a model that circles, and each wrapped agent runs in its own isolated runtime folder, so templates are never modified.

When at least one agent succeeded, the answer carries a **Create Flow** button. It turns exactly the successful tool calls into a canvas-loadable `.flw` workflow, normalized and secret-redacted by the backend, so a conversation can become a reusable workflow.

Step-by-Step
One concrete action at a time; she waits for your READY or output.

Ask Execs
A human Proceed/Deny gate before each risky step.

Exec Report
One table per agent family, one row per real tool call.

ACPX
Adds the external coding-agent and Skill tools to the surface.

◆ SELF-HEALING

Never hang, never discard, never lie

Every model call inside the loop goes through a per-request self-healing invoker that switches to a genuinely different tactic each time a transient failure strikes.

Each attempt runs under a watchdog. The shipped configuration allows 900 seconds, deliberately longer than the 600-second model-client timeout, so the watchdog never abandons a call the client would still finish; the Cancel button is polled every quarter of a second, so she never blocks forever. The ladder cycles through six tactics and keeps going up to 4,096 of them. Only you, with Cancel, stop her.

If the ladder is ever exhausted after agents already ran, she finishes gracefully from that real work: the Create Flow button and the Exec Report survive, and the answer says what happened.

Recovery is narrated live. Status lines such as “Tactic #3 … I will NOT hang” stream into the chat as they happen, and a SELF-HEALING NOTE opens the final answer.

1	`normal`	The full request again.
2	`retry`	The same tool-bound request; most blips clear here.
3	`patient retry`	Wait, then retry with extra patience.
4	`trim context`	Keep the system messages and the last twelve.
5	`minimal`	Keep the last six messages; tools stay bound.
6	`plain summary`	Last resort: summarize truthfully what was gathered.

◆ **Real bugs are not hidden**

Only transient failures — timeouts, resets, HTTP 429 and 5xx — trigger new tactics. A deterministic error such as a bad schema is raised at once so it gets fixed.

◆ COMPACT MODE

Every request fits the model it is sent to

A model whose real window cannot hold the complete request receives a compact one; a model that can hold it receives exactly the request it always did.

Ollama does not refuse an oversized request for a local model: it keeps the end and answers from that. A local qwen2.5 whose 32,768 tokens were split between two request slots read 16,386 tokens of a request of about 52,000, lost the live system metrics, and answered a CPU question with the time.

Ollama's own count is now the proof. When the characters sent cannot fit the tokens Ollama says it read, the request was cut, and that count is learned as the model's real window. A local model's reported context length is never taken as its per-request window, because parallel request slots divide it.

A request that fits the usable window is sent byte for byte, as before. Otherwise Compact mode keeps System-Metrics and Files-Search, whose context rides inside the question, and binds only the rows you tick. Since v1.75.0 that is a toolbar switch: ticking it unticks every Configure row but those three and pauses External MCPs, and each row shows its price. The system prompt is rebuilt by rule priority and closed by a seven-rule note, and history keeps the six newest messages. A first step that Ollama still cut is re-fitted and resent before any tool runs.

Full
A model that can hold the request receives it unchanged, byte for byte.

Compact
A toolbar switch: only the rows you tick are bound, each with its price.

Locked
A model too small for everything turns Compact on by itself and locks it.

Your choice
`context_compact_mode` is auto by default, or always or never, obeyed exactly.

◆ **It fails open to the full request**
Every part of the fitter fails open: when anything is uncertain, the complete request is sent exactly as before.

◆ MODEL BRAIN

Every model tuned from formal sources

Since v1.76.0 each model receives its own sampling, thinking level and context size, taken from formal sources, instead of one fixed set of values for every model.

The values come, in this order, from your own `model_brain_overrides`; a built-in knowledge base whose every profile cites its vendor model card or Hugging Face `generation_config.json` (GLM-5, DeepSeek-V4, MiniMax-M3, Gemma 4, Qwen 3.5, Qwen 2.5, gpt-oss, Kimi, Mistral Large 3 and Nemotron 3); a profile learned once, in the background, for a model she does not know yet and kept outside the installation so it survives updates; the parameters Ollama publishes for the model; and finally the model's own defaults. The requested context never exceeds what the model can really read.

Inside one answer, between tool calls, a thinking model gets its own earlier reasoning back, as its maker requires, instead of starting over at every step; it is dropped between your questions, and the context gauge counts it. In Config ▸ Models, Save opens an Auto-tuning dialog that tunes every configured model in front of you, with the source of each value.

The stall clocks that abandoned a slow model and started again from scratch were removed. Measured with Ollama's own numbers, reading even a very long prompt is not the bottleneck on Ollama cloud: the time is the reasoning the model generates, and a clock cannot tell a dead call from a model that is still thinking. The self-healing watchdog and the client time limit remain.

Formal sources
Your overrides, cited profiles, learned profiles, Ollama, model defaults.

Real context
The requested window never exceeds what the model can read.

Reasoning kept
Thinking models keep their train of thought between tool calls.

Auto-tuning
Config ▸ Models ▸ Save shows each model tuned, with its sources.

No stall clocks
A slow model is never thrown away and restarted.

Your choice
`model_brain: off` restores the legacy fixed sampler exactly.

◆ **No model name in the code**
A new model gets its values from the knowledge base, research, Ollama or its own defaults, never from a branch written for one model name, and a value you set always wins.

◆ EXEC REPORT

A verdict you can trust

An exit code is one bit. An agent's own structured self-report is a typed record, and it outranks the exit code.

`agent_verdict.py` parses each agent's `INI_SECTION` block into a typed tree and runs an ordered rule table over it. There is no model call and no heuristic, so the same input always yields the same colour. The status vocabulary is closed: five disjoint sets, united as `KNOWN_STATUSES`, and a repository-wide test rejects any status token an agent invents.

A read-only diagnostic that reports a problem has succeeded, because the finding is the deliverable. Degraded work, work not done and agent errors stay red. A red row therefore means that no clean requested deliverable exists, never merely that a check found something.

Rule	Fires when	Verdict
R1	The agent wrote no self-report	Exit code decides
R2	The agent declares error or failed	FAILED
R3	Work not done: refused, not_found, engine_unavailable …	FAILED
R3b	Degraded: tokens_only, compiled_with_errors, operator_required …	FAILED
R4	A diagnostic completed: invalid, findings, no_matches, listed …	SUCCESS
R5	An explicit success or ok flag	That flag
R6	A non-zero errors count	FAILED
R7	Nothing decisive and a non-zero exit code	FAILED
R7b	A named completion: ok, created, sent …	SUCCESS
R8b	A status token outside the vocabulary	SUCCESS, flagged
R8	No failure signal at all	SUCCESS

◆ **Why R4 outranks R5 and R6**

A linter that worked perfectly reports `status: invalid`, `success: False` and `errors: 2` together. The last two describe the document, not the agent.

◆ PERMISSION

Ask Execs: a human gate before risky steps

With Ask Execs on, the executor blocks before each risky tool and waits for Proceed or Deny in the browser. A single Deny halts the whole chain and leaves a red “Execution interrupted” banner.

The gate is an explicit allowlist, chosen tier by tier. Every doubtful path resolves to Deny: a failed prompt, a Cancel or a disconnected browser never lets a risky tool run unconfirmed. Unchecking Ask Execs during a run relaxes that run; checking it again re-arms the prompts.

Tier	Prompted	Agents
Command and script runners	Yes	Terminater, Executer, Pythonxer, SSHer, Kalier, Dockerer, Kuberneter, SQLer, Mongoxer, Gitter, Jenkinser, PSer, J-Decompiler
Editor persistence	Yes	Interactor new/save; reads and guided typing do not persist a file.
A · destroys or overwrites	Yes	Deleter, Mover, File-Creator, Editor, De-Compresser, unzip, PDFer, LaTeXer
D · remote systems	Yes	SCPer, Apirer, Nmapper, Discoverer, Crawler, NetSpeed-Calculator
B · messaging	No	Emailer, Whatsapper, Telegrammer, Instant Messaging Doctor: user authorization is still required
C · desktop and hardware	No	Keyboarder, Mouser, Windower, Playwrighter, firmware agents, Blenderer, Unrealer: visible while they act

◆ RETRIEVAL

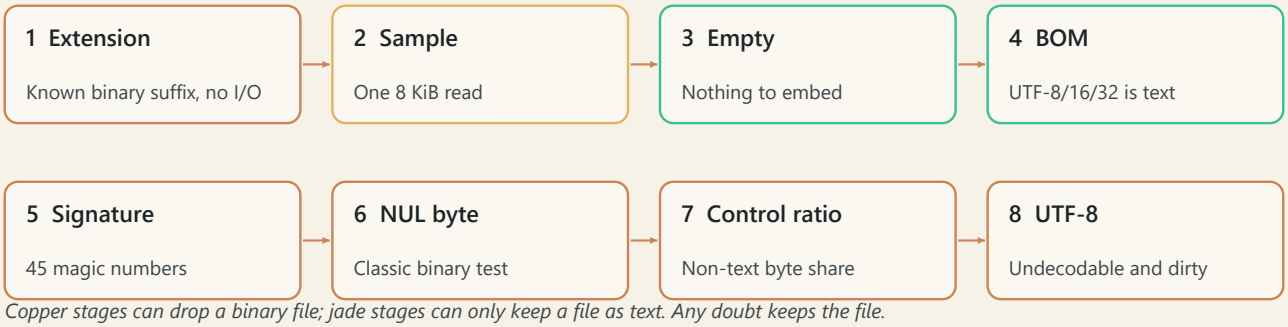
Grounded in your project, and only its text

Hybrid retrieval combines FAISS vectors with BM25 keywords, extracts metadata, budgets the context and falls back gracefully when a chain cannot retrieve.

When you load a directory or file, that loaded context outranks her self-knowledge: a request to “summarize the project” is answered from your project, not from a description of herself.

Before any file is embedded, a binary-content guard screens its bytes in a short-circuiting cascade that reads at most one 8 KiB block. Binary files are dropped exactly like a name you chose to omit, and every drop is named in `tlamatini.log` under `--- [BINARY-GUARD]`.

The guard fails open: any doubt means “load it as text”, because silently deleting real context is worse than embedding one odd file. The byte-order-mark stage runs before the NUL-byte stage, or every UTF-16 document would vanish.



Copper stages can drop a binary file; jade stages can only keep a file as text. Any doubt keeps the file.

◆ FLOW COMPILER

One compiler for the canvas and the chat

Canvas snapshots and chat-generated drafts compile through the same Agent Contract registry before any file reaches the session pool.

Each agent's contract declares which configuration field a connection on each slot writes, which fields Parametrizer may read, which paths are secrets to redact, and whether it is a singleton, long-running or never starts its targets. Both surfaces normalize into one `FlowSpec` (schema version 2).

The canvas Start button compiles the live snapshot in write mode, so an edited but unsaved flow runs under the same validation as a freshly loaded file. Validate uses a dry run that previews the compiled configuration without touching disk.

Agents that feed Parametrizer emit one atomic `INI_SECTION` block per result. 52 agents currently declare structured outputs.

```
INI_SECTION_APIRER<<<
url: https://example.org/status
status: 200

{"service": "ok"}
>>>END_SECTION_APIRER
```

◆ RELIABILITY

Built to keep running

Small, deliberate mechanisms keep the core alive under the conditions that used to stop it.

Console shield

The log file is written first; console output goes through a bounded queue, so clicking the console cannot freeze her. Since v1.76.0 lines are coloured by level.

Orphan reaper

Three tiers sweep dead descendants and orphaned console hosts after tools, after answers and at shutdown, without ever raising into the chat.

Per-user log lines

Lines carry a five-character tag such as [a3] so concurrent users and turns never blur.

Clean Ctrl+C

The signal handler only sets an event; a worker cleans up, a watchdog guarantees exit, and a second Ctrl+C exits at once.

Configurable port

`django_port` moves the web port with no rebuild and falls back to 8000 on any bad value.

Temp and Templates

Scratch files live under the application's Temp folder; scaffolded projects default to its Templates folder.

CHAPTER

III

Where You Work With Her

The chat, the two canvases, the catalog and the dialogs that operate everything else.

IN THIS CHAPTER

- From a magnet to verified files
- Choose how she types and teaches
- A command follows the learner's controls
- Hear the step before the first key
- Edit the document you are discussing
- Save into the project folder you choose
- Teach inside the live, unsaved document
- Immediate help while the learner types
- Conversation with modes
- More room to write, or more room to read
- Drop a message from the conversation
- What remains after Drop
- Speak beside Send
- Config ends with Mic
- One main console, no extra window
- Measured source behavior
- The menu bar in v1.92.4
- Read usage while she works
- Different scopes, clearly identified
- Report what the account supplies
- Design agent workflows visually
- Draw a conversation, then press Play
- Playback that stays in its own lane
- Writing and formatting a commentary
- Comments survive save, open and draft recovery
- The right editor, from chat or Windows
- Register, repair and remove both formats
- Remove the application and its owned workers
- Install and remove with a way back
- Repair the boundary, then repeat the real operation
- More room for the canvas or the log
- Ready-made prompts, grouped by purpose
- Read a PDF, then hand her the whole document
- Markdown opens as a page, and stays your source
- Drop a file anywhere, and she knows what it is
- Every model choice in one place
- Every page shares one theme and one way to resize
- Everything else, without editing files

◆ TORRENTER

From a magnet to verified files

Chat, the Agentic Control Panel and XT share the account's Jobs queue.

Inspect the metadata and select file IDs. A job receipt lets you reconnect. Enable Wait for complete download in a flow when downstream nodes need the files.

A fifth magnet can wait for a metadata slot. Reuse its inspection ID. The verified manifest identifies the delivered copy in Downloads or the edition's fallback folder.

◆ CONFIGURATION · v1.91.0

Choose how she types and teaches

Config → Live Editor & Terminater opens one shared dialog on the chat and control panels.

Live Editor controls 40–250 WPM, a 0–3 second pause after each segment, teaching detail, field suggestions, ghost text versus choices, and following the typed cursor. Manual scrolling suspends following for that segment. Font, line numbers and wrapping affect the real editor. Teaching detail instructs the model; exact wording still varies.

Standard and Guided learning presets populate a draft. Preview types and speaks a harmless sample without editing files or executing commands. Save applies behavior to the next action. Cancel, Escape and Close discard the draft. Stop, mute and current-tab editor access are immediate.

Pace
Separate speeds or a linked 40–250 WPM setting; no instant typing.

Teaching
Brief, balanced or detailed guidance with optional local suggestions.

Appearance
Editor and terminal presentation saved per account/browser.

Account
Behavior persists on the server; concurrent saves detect revision conflicts.

◆ TERMINATER SETTINGS

A command follows the learner's controls

One TerminalHub resolves account preferences across chat, ACP, Prompt Flow and browser actions.

Choose 20–250 WPM, an available default shell, advisor for new terminals, a command wait window, a no-output threshold and a user handoff wait. Shell and advisor defaults affect new terminals. A wait timeout does not kill the process. Layout, font and cursor changes apply to existing views; resized prompts retain honest screen-based read-back.

I always press Enter prevents automated submission, including aliases, encoded text and control characters; the user's keyboard still works. New flows can use account defaults. Older flow files retain explicit values. Manual Enter and narration remain account choices, and typed chat preserves a shell explicitly requested by the task.

Type
One key at a time, 20–250 WPM; no paste or instant mode.

Submit
Task-controlled execution or user-only Enter enforced by the hub.

Wait
Command, no-output and handoff windows have separate meanings.

Inherit
use_my_defaults applies account preferences without rewriting older flows.

◆ COMMAND AND LESSON VOICE

Hear the step before the first key

Narration uses the existing browser voice and a shared, cancellable speech queue.

Enable narration independently for Terminater commands and Live Editor segments. Choose an explanation, literal command/code or both, spoken before, during or after typing. When waiting before typing is enabled, a real speech-completion receipt or explicit skip is required. An unavailable voice offers Continue without voice or Stop.

A single focused account tab speaks each terminal event. Receipts belong to that exact connection; disconnect, Stop and the bounded deadline can cancel before any key. A reconnect never replays a command. Secret-marked text and user handoff values are excluded. Outcome notices describe command-action status without reading terminal output aloud.

Shared voice
Reuse voice choice, rate, volume and pitch from Config → Voice.

Priority
Global mute and microphone listening override narration.

Timing
Before/during/after; optional completion before typing.

Control
Escape stops speech; Ctrl+Shift+M toggles global mute.

◆ CANVAS EDITOR · v1.89.0

Edit the document you are discussing

A local CodeMirror 6 editor keeps text, code, JSON and Markdown Source editable in the chat canvas.

Open or drop a file, make a change, and use the filename bar: Save, Save as, Undo, Redo, Find/replace and Wrap. Copy and Use as context take the edited source. Ctrl/Cmd+S saves; Ctrl/Cmd+Shift+S chooses a destination. Tab normally navigates; Ctrl+[and Ctrl+] indent.

Markdown Preview is sanitized with the existing Marked/DOMPurify renderer. Source and Preview share one buffer and undo history. HTML and SVG remain text, never an active web page. PDFs remain read-only viewers and export their original bytes; flows and images keep their own routes.

The dirty marker survives failed or cancelled saves. Open, Clear and Refresh ask before discarding edits. Unsaved text lives in memory, so save before closing. Automatic sizing reserves 320 CSS pixels for the canvas and wraps controls; manual splits remain the user's choice.

One document
Code, configuration and plain text share the same editing authority.

Markdown
Editable source and sanitized preview preserve undo history.

Controls
Save, Save as, Undo/Redo, Find/replace and Wrap.

Layout
A reserved canvas width and an editor that fills its available height.

◆ NATIVE SAVING

Save into the project folder you choose

The local native Windows picker makes Downloads optional.

Save uses a browser-granted writable handle or an already authorized native destination. Without one, it asks. Save as and the navbar Save open a native Windows Save As dialog on a local connection; subsequent toolbar Save writes there and Refresh reads it. PDF exports stay copies. Remote browsers need their own supported File System Access picker.

A complete revision check refuses external changes. Writes stage on the destination volume, flush and recheck before replacement. ReplaceFileW preserves existing Windows security, named streams and creation time; partial failure retains a recoverable original. The protected database backup/restore path is unchanged.

Unedited files keep exact bytes. Edited files retain encoding/BOM and existing line endings; new lines use the original first terminator. UTF-8 and BOM-marked UTF-16 are decoded strictly. Malformed/binary data is refused. Text editing and native saving are limited to 64 MiB.

Choose
Native Save As to any writable local folder.

Authorize
Account/session-bound capability; 12 idle hours, lost on restart.

Compare
External disk changes refuse overwrite; Save As keeps a separate copy.

Preserve
Encoding, BOM, line endings and existing Windows metadata.

◆ INTERACTOR · v1.90.0

Teach inside the live, unsaved document

Interactor is a module and a built-in tool, not a workflow agent or a terminal.

Enable Multi-Turn, Interactor in Configure MCPs/Tools and Live editor above the canvas. Ask: Help me create a Dockerfile for a Java servlet application using your text editor; teach one step at a time. The English/Spanish editor aliases are taught in prompt.pmt.

First read the exact current buffer, selection, document ID and revision. With no requested path, new creates an empty file under Temp/interactor/<account>/<random>/ using the requested filename. Existing files are never overwritten. Tlamatini types a short known segment, then guide hands the next field to the learner and prevents further mutations in that turn.

Continue sends a normal new chat request and reads again before reviewing the learner's answer. Another hint and Explain selection also use normal chat without replacing an existing draft. Typing does not save or execute code. The ordinary Use as context snapshot is separate from Interactor's fresh read of unsaved text.

Read
Exact unsaved text and selection, never a stale disk substitute.

Type
One small visible teaching segment; one undoable edit.

Hand over
Explain the learner's next field, then stop tool mutations.

Review
Continue starts a fresh read in a new chat turn.

◆ TUTOR COMPLETION

Immediate help while the learner types

The model plans the next field once; local matching avoids a model request on every keystroke.

Ghost text appears only for a matching prefix among bounded choices prepared for the current field. Tab accepts, Ctrl+Right accepts a word and Escape dismisses. Suggestions are optional; the learner can write an alternative value. This is contextual teaching support, not a claim of unrestricted predictive autocomplete or vendor benchmark latency.

User input, selection, Stop, Escape, chat cancellation and disconnect interrupt AI typing. The typed prefix stays editable. A temporarily hidden or occluded tab pauses the same action for up to two extra minutes and resumes at the exact next character when visible. Timeout never replays a partial edit. Switching Markdown to Preview or hiding the canvas returns control.

Live tutoring accepts at most 262,144 UTF-16 units and checks document/revision before mutations. A smaller configured tool-result/context budget can refuse the receipt. Oversized content is refused explicitly, never truncated into misleading context.

Local matching
Prepared field choices; no model call per key.

Optional
Accept a suggestion, take one word, dismiss it or type your own value.

User priority
Interruptions preserve the typed prefix and invalidate stale work.

Honest latency
New questions still wait on chat/model processing.

◆ THE CHAT PAGE

Conversation with modes

The chat page is where you talk to her. Up to eight toolbar switches decide how each request runs.

Beside the conversation sits a canvas that shows code, text and PDFs. Paste a screenshot with Ctrl+V or drop images on any page: she saves them to her Temp folder and writes the full path into your message, ready for Image-Interpreter. A context ring between the toolbar and the message box shows how much of the model's window the real request uses, measured by the backend in bytes. Tokens use Ollama's own prompt_eval_count for the request; an estimate is labeled until a measured count arrives. When Ollama cut a request the ring says so, and an answer built from a cut request carries a CONTEXT-WINDOW warning.

Her avatar speaks answers aloud when you ask. Since v1.72.3, clearer syllable movement and faster closure make her lips easier to follow, with opening capped to her portrait. After login, pressing Enter on the welcome page takes you straight to the chat.

Answer tables stay readable with every model: only a cell whose text would fall below 3:1 contrast is recoloured, while readable tables, Exec Report tables and gradient backgrounds keep their colours.

Switch	What it does
Multi-Turn	Planned, tool-calling operator mode; unchecked means direct one-shot answers.
Exec Report	Appends per-agent execution tables; available while Multi-Turn is on.
ACPX	Adds the twelve ACPX and Skill tools to the bound surface; off by default.
Ask Execs	Proceed/Deny before each risky tool; a Deny stops the chain.
Step-by-Step	One concrete action at a time, waiting for your reply before the next.
Internet	Allows a web search to add context to the answer.
Compact mode	Binds only the rows you tick; locked on when the model cannot hold all.
Self-modify	Source runs and self-modify builds; sends self-knowledge, locked off if too big.

◆ THE MESSAGE BOX

More room to write, or more room to read

Drag the divider above the message box. With her avatar undocked, the box can shrink to its buttons.

The divider between the conversation and the message box follows the pointer and the keyboard. The arrow keys move it one step and Shift+arrows five; Home or a double-click shrinks the box to its smallest, End makes it as tall as allowed, and Escape ends a drag. Screen readers hear the box height. The toolbar, context gauge, attachment and dictation rows are measured, so none of them is cut off.

Docked, her portrait keeps its room above the action buttons. Undocked, the box can shrink to those buttons plus a six-pixel guard, and docking her again gives the portrait its room back. In a short window the box scrolls inside itself, and on a narrow screen the workspace pans while the page and its dialogs stay in place.

Docked, her portrait now follows the divider too: dragging the bar down shrinks it to a 120-pixel square and dragging up gives it back. Shipped in v1.83.0.

In v1.85.0, the portrait and its frame stay centered over the complete Mic-to-Send row. Views → Canvas can give the chat the full width; Views → Reset layout restores both panes and docks the portrait while keeping the document, draft and context.

Drag or type
Pointer, arrows, Shift+arrows, Home and End move the divider.

Undocked
The message box can shrink to its own action buttons.

Docked
A centered portrait above Mic through Send, down to a 120-pixel square.

Small windows
The box scrolls itself; dialogs stay inside the window.

◆ CHAT HISTORY

Drop a message from the conversation

Every message card has Drop beside Copy, for your prompts and Tlamatini's answers.

Press Drop and read the themed confirmation. Cancel starts focused, so Enter cancels by default. The red Drop button confirms deletion. Escape and the close button also cancel. The saved message disappears from the chat and the database history used for future requests.

The other messages stay. Dropping only a prompt leaves its answer; dropping only an answer leaves its prompt. Remove both cards to remove the exchange. A user prompt also takes its directly following Referenced Rephrase rows, so a reload cannot show those duplicate words.

Connected tabs for the same user remove the saved cards after the server confirms. A status, error or retry card without a saved message id only disappears from the current screen. The connection refuses Drop while it is answering, and a disconnected chat asks you to reconnect.

Choose
Drop sits beside Copy on both user and assistant message cards.

Confirm
Cancel starts focused. Red Drop confirms; Escape and close cancel.

Remove
Delete the saved message from chat history. Its paired prompt or answer stays.

Continue
The next request reads the updated history without a reconnect.

◆ MEMORY AND EFFECTS

What remains after Drop

Deleting a chat card changes future conversation context. Separate saved notes and completed work remain.

Each request reloads up to eight newest AgentMessage rows through DBChatHistoryLoader, then filters status and rephrase content. No cached conversation summary or LangGraph checkpoint needs clearing. Removing a row lets the window reach an older row when one exists.

An enabled External MCP memory tool stores notes separately from AgentMessage. Drop does not erase that knowledge graph; ask Tlamatini to forget the saved note too. Retained messages may also repeat a fact from a deleted card, so Drop is not a guarantee of complete forgetting.

Files created, messages sent, agent runs and Exec Report rows remain real completed work. Dropping their prompt or answer cannot undo those effects. A Create Flow button belongs to its answer card and disappears with it, so download a needed .flw before deleting the card.

History
Every request reloads a window of up to eight newest database rows.

Saved notes
External MCP memory is separate. Ask her to forget its note too.

Other messages
The paired answer or prompt stays and may repeat the same information.

Completed work
Files and sent messages remain. Save a needed flow before dropping its card.

◆ DIRECT VOICE INPUT

Speak beside Send

The microphone captures first; no model tool-selection step stands between the click and recording.

Click Mic beside Send, wait for Listening, and speak into the microphone on the Tlamatini host. Real input level, elapsed time and the silence countdown stay in the chat. After the configured silence window (shipped default 3.5 seconds), recognition runs and the text is appended to the existing draft. Config > Mic selects automatic normal-form submission or an editable draft for manual Send.

Review mode starts no task until Send. A second click or Escape cancels; empty or failed recognition sends nothing. Current modes and permission behavior remain in force. The avatar acknowledges dispatch through browser speech and respects Silent mode.

Click
Direct Whisperer start; the host microphone opens only for a recording.

Listen
Live samples drive the in-chat indicator and the configured silence gate.

Recognize
Local faster-whisper or the explicitly configured cloud speech engine.

Send or review
Submit once with current modes, or review the editable draft and press Send yourself.

◆ CONFIGURATION

Config ends with Mic

Choose automatic Send or an editable draft, using the same dialog style as the rest of Tlamatini.

The button label is Mic. Config > Mic follows Voice and controls what happens after recognition. Send automatically is the default. Keep in the chat input restores editing without launching a task or processing acknowledgment; correct the words, then press Send.

Preferences are saved in this browser for the next recording. Each recording owns a snapshot. Unset fields inherit the current Whisperer template; Reset then Save restores that inheritance. Cancel, close and Escape discard unsaved edits; outside clicks leave the dialog open. Config > Voice controls avatar playback; Config > Models > Speech selects the recognizer.

Input
Configured/default/listed host device, refresh and software gain from 0 to 300%.

Sound gate
Silence 0.3-20 seconds, cap 5-600 seconds, adaptive or manual sensitivity.

Recognition
Language, English translation, capture rate/channels, local beam size and VAD.

Guardrails
Validate settings before capture; removed or ambiguous saved devices cannot silently change inputs.

◆ INTERNAL SERVICE

One main console, no extra window

Developer visibility requirements do not become additional product windows.

The authenticated same-origin /ws/chat-voice/ connection prepares one resident worker per web process. Source/carried Python runs chat_worker.py; the frozen Django process stays free of the speech ML stack. A private token-authenticated loopback socket carries control and result frames. Preparation does not open the microphone.

No PowerShell, conhost, focus activation or console-visibility prerequisite is involved. The worker's stdout/stderr flows through agent.chat_voice_runtime into the main console/log. Failed startup and shutdown reap the child. Config reloads per job; compatible models are cached, and late cancelled results cannot submit.

UI ownership
The composer displays recording and transcription; no second console.

Configuration
Model/template inheritance plus validated, per-recording Mic capture overrides.

Boundary
The direct button is input, not a wrapped tool or an Exec Report row.

Release
Ship consumer/runtime/settings modules, carried validator and worker, Mic JS/CSS and template.

◆ VERIFICATION SCOPE

Measured source behavior

Dated source checks are evidence, not a claim that an older installer contains the change.

On September 27, 2026, 108 voice regressions and 42 existing dialog-theme tests passed. The worker refreshed 16 host inputs and restarted twice without extra windows or focus changes. Earlier real host capture yielded samples in 297 ms after direct start; cancellation and normal worker exit were observed. Two startup/shutdown cycles created no additional visible windows and left foreground focus unchanged.

The updated headed-browser suite passed 62 controlled-transport checks, including both modes, settings persistence, shared control styles and narrow layouts. Earlier cached recognition of a 5.768-second synthetic speech file took 0.203 seconds, versus 11.094 seconds on first decode. These are separate observations, not end-to-end latency guarantees. No new live Ollama task or rebuilt installer was certified by those checks.

Current source
108 voice regressions, 42 theme tests and real worker metadata/lifecycle probes.

Browser evidence
62 headed checks with controlled transport; separate from live ASR.

Limits
Cold model loading, provider latency and hardware vary; no zero-latency promise.

Detailed guide
docs/chat-microphone-design.md records configuration, protocol and evidence.

◆ NAVIGATION

The menu bar in v1.92.4

Views controls the workspace, Panels opens the flow editors, and Config holds settings.

Menu	What you find there
Open · Save	Open a canvas file; menu Save chooses a destination with native Save As.
Views	Show Canvas or Terminals; Reset layout restores the workspace.
Context	Load a directory or file as context, and set file-type omissions.
Panels	Agentic Control Panel, Prompt Flow Panel and, for staff, the Admin Panel. It stays usable while she is busy.
ACPX-Skills	Browse, configure, diagnose and reload the skill catalog.
External	The External MCPs dialog: catalog, activation and runtime readiness.
Config	Configure MCPs, Configure Agents, Models, URLs, Contacts, Access Keys Wizard, Voice, Mic, and Live Editor & Terminater (pacing, narration and interaction preferences).
DB	WAL-safe database backup and staged replacement.
Reconnect · About	Rebuild the chat connection; version, author credits and Check for updates. Since v1.78.0, About also opens Usage, even while she is busy.

◆ ABOUT → USAGE · NEW IN v1.78.0

Read usage while she works

ABOUT USAGE follows the shared dialog theme and remains available during a long answer.

Open About → Usage, then choose Cloud credits, Chat activity or Models. The first tab reads the configured Ollama account; the other views identify measured main-chat activity for the signed-in Tlamatini user. Recording begins with this feature and does not reconstruct earlier history or include child agents and other applications.

The desktop dialog is 80vw by 74dvh, capped at 1440 pixels wide: at most 59.2% of the client area. Narrow screens use 94vw by 63dvh. Text follows normal dialog sizing, buttons have clear borders and the body scrolls between fixed controls. Fractional displays have at most two decimal places. Colored bars offer keyboard-focus and pointer tooltips plus exact tables.

Cloud credits
Reported balance, monthly usage, allowance and refill weeks/date.

Chat activity
Your measured calls, input/output tokens and daily trends.

Models
Recorded model activity and peak UTC dates; server inventory alongside it.

While busy
Open and refresh without interrupting the model request.

Normal size
Under 60% client area; ordinary typography and clear buttons.

Accessible
Keyboard tabs, chart tooltips, focus trap and normal Escape/Close behavior.

◆ ACCOUNT DATA

Different scopes, clearly identified

Monthly credits, rolling cloud usage and recorded local activity answer different questions.

The authoritative /api/balance response supplies included allowance and remaining balance, purchased credits and the UTC refill period. Used credits equal allowance minus included balance; total available adds purchased credits only when both values are reported. Plan names never supply an assumed allowance. Refill weeks/date come from the returned period, not an assumed 30-day cycle.

Cloud /api/usage supplies rolling 7/30-day history, including today's partial UTC day. That spend is not monthly credits used. Cached input is already part of input. The local ledger's 7/30 calendar dates include today and have a narrower measurement scope.

Ollama's documented API does not yet expose the website's full per-model history or individual purchased-credit expiries. The dialog links to Ollama Usage for website-only details. Missing fields are explained, never replaced with invented values. Balances follow the configured server account, which can differ from the user's website login.

Monthly balance
Actual included and purchased credits from /api/balance.

Rolling history
7/30-day cloud spend and tokens; separate from the billing period.

Local model graphs
Measured Tlamatini main-chat activity, not website-wide history.

Refresh
Healthy data is cached 60 s; a brief outage is retried after 15 s.

Missing data
Explain unavailable or stale values; no guessed balance or token count.

References
docs.ollama.com/api/balance and /api/cloud-usage; docs/usage-dashboard.md.

◆ ADAPTIVE LAYOUT

Report what the account supplies

The response shape selects the layout; a plan label is not a credit entitlement.

Free, Pro and Max accounts with monthly balance fields use credit cards. Legacy session and weekly responses use reported remaining percentages and reset times; unsupported cost and token fields are omitted. Purchased-only responses display only the supplied balance. A zero monthly allowance does not produce an invented percentage.

As checked on October 10, 2026, Ollama lists Free, Pro, Max, Team and Enterprise, with no cloud plan named Unlimited. Unlimited local execution is a different claim. Team balances are shared, while this adapter requests default member usage, not administrator team scope. Commercial terms remain at ollama.com/pricing; they are not hardcoded into the dialog.

Monthly
Returned allowance, balance, purchased credits and reset period.

Legacy
Remaining session/weekly percentages; no guessed dollar conversion.

Purchased only
Only the reported purchased balance; no invented included quota.

Free or zero
Report actual data and omit percentages when allowance is zero.

Team
Shared credit balance is distinct from member-scoped usage.

Unlimited label
Never interpreted as unlimited cloud credits.

◆ AGENTIC CONTROL PANEL

Design agent workflows visually

Drag any of the 87 agent types onto the canvas, connect output triangles to input triangles, configure each node, validate, and press Start.

Agents run as separate, log-emitting processes that talk through their `config.yaml` files, with a live LED for each. Flows can be paused, resumed and stopped, and saved or loaded as `.flw` files. FlowCreator can design a flow from a sentence, and FlowHypervisor watches a running flow and raises an alert when something looks wrong.

Since v1.70.0 the canvas is a real editor. Duplicate copies agents together with their saved settings and the connections between them; Undo also restores an agent's configuration. Editing locks while a flow runs, so a flow is never changed underneath its own processes.

Undo and Redo
Up to 1,024 steps, configuration included.

Duplicate
Copies settings and internal connections.

Zoom
25 to 200 percent, and Fit.

Agent search
Filters the 87 agent types by name.

Starters
Locates each Starter on a large canvas.

Help
Explains every gesture; • marks unsaved changes.

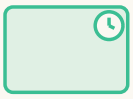
◆ PROMPT FLOW PANEL

Draw a conversation, then press Play

The Prompt Flow Panel turns a chain of prompts into a diagram of eight executable operations plus static review notes, saved as a portable `.fpmt` file and played against your configured models.



Prompt
Sends its text to the model; Multi-Turn and ACPX can be enabled.



Programmed Prompt
Waits for a scheduled time or a delay, then sends.



Decision
Takes Yes or No from the last answer, or asks you.



Feed embeddings
Adds reference text to this run's retrieval context.



Flush embeddings
Clears the run's retrieval context.



Clean History
Clears the run's conversation and last output.



User Input
Stops to ask you something; your reply joins the run.



Terminater
Types commands in a real visible terminal and reads their results.



User Commentary
A static speech-bubble note for reviewers; it never runs.

◆ PROMPT FLOW PANEL

Playback that stays in its own lane

Validate checks the diagram, Play runs it, and every run keeps its own conversation, embeddings and cancellation identity, so it never disturbs your chat.

Write `{{last_output}}` to pass the previous answer forward. Decisions compare strings — contains, does not contain, equals, empty, with optional case matching — and never evaluate expressions from a file. Cycles are allowed and bounded by a maximum number of executed operations, 500 by default and up to 5,000.

The running figure lights up and traversed connections are highlighted. Pause lets the current operation finish and holds the next; Stop drains the running worker before another run may start. Opening a file never runs it, and a User Commentary note never runs at all.

Feed embeddings requires a real vector store. A provider error cannot become success through the chat stack's prompt-only fallback, including with tools enabled. The flow reports failure before its next operation and keeps any previously accepted context.

Format
JSON `tlamatini-prompting-flow`, version 2; version 1 files migrate.

Limits
5 MiB, 500 assets, 1,000 connections.

Isolation
Empty history and embeddings at every start.

Validation
Checked in the browser and again on the server.

Not .pmt
Plain-text system prompts keep .pmt and are refused.

Drafts
Kept per user in browser storage.

◆ STATIC REVIEW NOTES

Writing and formatting a commentary

User Commentary is a review note on the canvas. User Input is the operation that pauses playback for a reply, using the notched figure.

Double-click a bubble or press Enter to write directly in it. Select words or paragraphs, then use the floating mini toolbar for font, size, text color, bold, italic and underline. For example, one bubble can have a large Verdana heading, an italic Arial sentence and a Georgia paragraph. With only a caret, formatting applies to subsequent typing. Outside editing, it applies to the whole selected note. Bubble color and alignment affect the note.

Drag any of the eight borders or corners to resize, even while writing. The handles also accept arrow keys. Fit text removes spare height, and the bubble grows to contain all wrapped text. There are no internal scrollbars, numeric size fields or commentary configuration dialogs.

Done or Ctrl+Enter saves the complete edit. Cancel or Escape restores its original text, styles, dimensions and position. Ctrl+Z/Y works inside the editor. Completed edits, moves, resizes and duplicates participate in the diagram's Undo/Redo. Copying between comment editors preserves formatting. External clipboard text stays literal.

◆ PORTABLE FLOW FILES

Comments survive save, open and draft recovery

Version 2 .fpmt files retain independent notes with their full text, formatting and geometry.

Each note stores allowlisted text runs plus matching literal text. Runs carry font, size, text color, bold, italic and underline. Older plain version 2 notes normalize to one run. Version 1 executable commentary migrates to User Input with its identifiers, connections, settings and reply/cancellation behavior intact. Opening a file never starts playback.

A note has no connection ports or Start state and never enters model context, conversation history or playback step counts. Tokens such as {{last_output}} and HTML-like text remain literal in a comment. Several independently styled notes can share a diagram.

The per-user browser draft keeps the existing .draft.v1 key for compatibility. Portable files retain the .fpmt extension. The similarly named .pmt files belong to system prompts and remain a separate format. Limits are 5 MiB per flow, 500 assets, 1,000 connections, 100,000 characters and 10,000 formatted runs per note.

◆ OPENING FLOW FILES

The right editor, from chat or Windows

The shipped file associations open .flw agent flows and .fpmt prompting flows. Opening never runs them.

The main chat routes both formats into their own editor tabs and keeps its current document. An external file launch validates a bounded snapshot and reuses a matching running server before any database startup work. A cold launch starts the application on its configured port.

Login returns to the requested file. Short-lived, single-use opening links carry a random token rather than a filesystem path; browser uploads are bound to the signed-in user. Invalid formats, unsupported versions and files over 5 MiB are rejected before replacement.

Unsaved diagrams and recovered drafts require confirmation. Rich commentary runs retain their fonts, styles and layout through .fpmt save/open. Agent-flow opening now awaits validation and session preparation; failed credential redaction blocks unsafe .flw downloads.

From chat
Open either flow in its editor and keep the chat document.

From Windows
Start once or reuse the running app; return through login.

Preserve work
Validate before replacement; confirm unsaved changes.

No automatic run
Opening prepares the diagram. Play remains a user action.

◆ WINDOWS FILE TYPES

Register, repair and remove both formats

Per-user associations, installation ownership and Windows default-app choices.

A shared PowerShell helper registers .flw and .fpmt with separate ProgIDs, friendly names, icons, Open With entries and Default Apps capabilities. Status reports the current registration; DefaultApps opens Windows Settings so the user can choose a default.

Installation registers both formats. Update Repair preserves explicit unregistration and never takes another installation's owned entry. Removal deletes only this installation's values, preserving Windows UserChoice and other applications' Open With entries. Historical .flw wrappers cover both formats so existing installer/uninstaller binaries remain compatible.

The scripts and opening modules are included in frozen and self-modify packaging. Native registry checks use an isolated test key, and visible Chrome tests exercise real file opening and login. The October 4 release campaign also rebuilt local main and uninstaller executables. The work was committed to main that day and is carried by the v1.76.0 tag; compiled acceptance is recorded separately from source tests. See docs/windows-flow-files.md for commands, contracts and verification.

Install
Register both types with names, icons and the correct editor destinations.

Administer
Inspect Status, register or remove one type, or open Default Apps.

Update
Repair owned entries without undoing the user's opt-out.

Uninstall
Remove owned registrations; keep other apps and user documents.

◆ WINDOWS LIFECYCLE

Remove the application and its owned workers

Confirmation, installation ownership, retryable errors and preserved user content.

The real installer registration failure came from an omitted InstallDir whose parameter default evaluated before Windows PowerShell populated PSScriptRoot. The wrappers now resolve that default inside the script. Fresh powershell.exe -File tests exercise the same entry point from an unrelated working directory.

The uninstaller first requires the main application to close. After confirmation it stops workers belonging to the selected installation and their descendants. Exact path boundaries, creation times and inherited TLAMATINI_AGENTS_ROOT identify agent children even after reparenting; unrelated processes and Explorer remain running.

A temporary independent copy allows the installed uninstaller itself to be removed. Helpers and the installation marker remain until file and owned-registry removal succeed, so a partial failure can be retried. Locked files and denied registry deletion report failure. Drive roots, shared folders, source checkouts and redirected targets are rejected.

Since v1.76.0, when Windows still holds Uninstaller.exe itself, the file is retried, moved aside and removed by a hidden cleanup after the uninstaller exits, so the uninstall finishes cleanly.

Agents and nonempty user-content directories remain under the existing preservation contract. A full directory/registry reset is a separate authorized action. Verification distinguishes real compiled GUI outcomes from backend tests and records any blocked native launch explicitly.

Installer entry points
Resolve the installation directory inside PowerShell script bodies.

Owned workers
Stop this installation's agents and surviving descendants after confirmation.

Honest retry
Retain removal support until success; report locked files and registry errors.

Preserve user work
Keep agents and nonempty content; notify the shell without restarting Explorer.

◆ SETUP WINDOWS · v1.82.0

Install and remove with a way back

The Installer and Uninstaller share one window that prepares a recovery copy before changing anything.

Both executables use one setup window: Tlamatini's palette, keyboard controls, a destination and review page, real progress and result details that stay on screen. The action row is its own window row, so buttons are never clipped, and large fonts or long paths scroll the body. Errors stay in the window with Retry and Change folder, and every run writes a private log under %LOCALAPPDATA%\Tlamatini\SetupLogs.

Before installation or removal changes a file, a recovery copy is prepared and verified; its journal lives in the installation's Temp folder, and the next run restores an interrupted transaction first. A failed restoration is reported, never presented as a rollback. A locked file gets a five-second retry and is then named with the program holding it, which is never closed for you. Only Tlamatini's own registry values and Desktop shortcut are snapshotted.

These are custom EXE recovery mechanics, not MSI transactions. In the visible October 11-12 campaign 52 interface checkpoints, 28 recovery checks, 21 lifecycle checks and 61 mechanics tests passed; five real installations and four real removals passed their independent audits. The fifth removal stopped safely before deleting anything, because a test browser held a file in the installation's Temp folder.

One window
Shared Installer and Uninstaller screens that never clip their buttons.

Recovery copy
Prepared and verified before any file changes; restored after an interruption.

Named locks
A locked file and its holder are named; nothing is force-closed.

Own scope
Only Tlamatini's registry values and shortcut; never UserChoice.

Bounded helpers
PowerShell helpers end within 120 seconds inside a Windows job.

Honest tally
Five installations and four removals passed; one stopped safely.

◆ LOCAL RELEASE VALIDATION

Repair the boundary, then repeat the real operation

The October 4 repairs, now carried by v1.76.0, have separate source, frozen, package and visual evidence.

The visible campaign checks the actual installer, Windows associations, authenticated file opening, administration, main chat and both flow panels. It also compares several rich comments across repeated file writes and reads, executes local agents, and audits owned workers after shutdown. A configuration dialog opening is not proof that a hardware or external-service agent executed successfully.

System-Metrics and Files-Search now honor the effective configuration and nondefault ports, including UTF-8 BOM files. The path-security and chat-chain readers accept the same encoding without weakening directory boundaries. Both auxiliary context chains use the same effective paths and endpoints; system sockets close after each request and file RPCs have deadlines. Parametrizer restores connections when a saved file contains mappings without explicit connection lists. Model-free agents avoid unrelated model-setting reads. Numeric messaging identifiers keep the same string/integer contract in public and keyed templates.

A rejected embedding request must fail the flow before its next operation. The campaign's provider returned HTTP 401; this is recorded as a blocked capability with a tested failure path. The desktop-control tool also blocked the compiled uninstaller launch. Backend cleanup tests do not certify its unobserved native confirmation and completion screens.

Detailed results and retained failed attempts are in docs/changes/2026-10-04-release-validation.md. The review profile requested by Angela lists each changed file and code segment. The work was committed on October 4 and is carried by the v1.76.0 tag.

Real user paths
Installer, login, Admin, chat, both panels and Windows file opening.

File fidelity
Mixed styles, geometry and rendered text checked across save/open cycles.

Runtime boundaries
Configured MCP endpoints, agent defaults, mappings and identifier types.

Explicit limits
Provider failure and blocked native launch remain visible in the verdict.

◆ RUN OUTPUT

More room for the canvas or the log

Drag the horizontal divider above Run output to give it 5% to 95% of the available pane height.

The percentage uses the combined canvas and output height, excluding headers, the status strip and the divider. The initial allocation is 20%. The canvas and Run output each keep their own scrollbar. Moving the divider preserves canvas zoom, node dimensions and text size. The Operations bar keeps its separate horizontal sizing control.

Tab to the divider and use Up/Down for one percentage point, or Shift+Up/Down for five. Home selects 5% and End selects 95%. Escape ends a drag. Click the Run output heading to collapse it. Reopening restores the chosen height, and resizing the window retains the ratio.

The browser remembers the output height per signed-in user. This layout preference does not modify a flow file, mark the diagram as edited or add an Undo step. Resizing the panel also leaves the run and its output unchanged.

◆ CATALOG OF PROMPTS

Ready-made prompts, grouped by purpose

The Catalog of Prompts opens with 16 sections, each beginning with a guided Step-by-Step wizard and growing from simple to advanced.

Sections: Voice Commands, Getting Started, Files & Search, Run & Execute, Code & Project Generation, Documents & PDF, Images & Vision, Agents & Flows, ACPX, Skills & MCPs, Desktop Automation, Games & 3D, Firmware & IoT, Security & Recon, Messaging & Contacts, Media & Voice, More. VOICE COMMANDS comes first, because speaking is the shortest way to use her: you talk, Whisperer transcribes, and the transcript becomes the prompt. These catalog workflows use the model-mediated wrapped tool and their card-specific modes; they are distinct from the direct microphone button beside Send.

The mode badges on each card are derived from the prompt text itself, and clicking a card sets the toolbar switches to exactly those modes. Every prompt uses one grammar, so it is always clear whose blank is whose.

Mark	Filled by	Meaning
[[…]]	You	A value you provide, collected at the top with a safe default.
{{ … }}	Tlamatini	A value she fills at run time.
< … >	Report slot	Where an answer is printed; never an input.

◆ PDF CANVAS

Read a PDF, then hand her the whole document

Open a PDF beside the chat in a vendored Mozilla PDF.js viewer, then use it as context.

Reading uploads nothing and calls no model: the viewer reads byte ranges straight from your disk, so there is no size or page cutoff on Tlamatini's side. Text selection, navigation, zoom, fit, rotation and in-viewer passwords behave as you expect; Copy lifts the text of every page and Save As returns the original bytes.

Use as context asks one question first: Process images? It starts unticked every time, and nothing runs until you press Continue. Unticked, she reads the selectable text only. Ticked, she also renders every page and analyzes every embedded image with Image-Interpreter. Progress is honest, Cancel really cancels, and partial image failures are reported, not hidden. A PDF password is used in memory only: never written to disk, to the context text or to the log.

A second account signed in from the same browser takes over every open tab, while an open chat keeps its first account, so loading is refused. Since v1.76.0 the message names that cause and the fix, reloading with F5, the header comes back and the progress dialog marks the failed step.

◆ MARKDOWN CANVAS

Markdown opens as a page, and stays your source

Open or drop a .md file and the canvas renders it; Source opens the editable document.

Open, Refresh, a dropped file, a generated-file link and Context ▸ Set file as context all recognise .md and .markdown. Preview renders CommonMark and GitHub-flavored Markdown: headings, tables, task lists, fenced code through the existing highlighter and safe embedded HTML such as expandable details. Source edits the current text with line numbers.

Rendering is local: Marked 18.1.0 and DOMPurify 3.4.16 are vendored with pinned integrity, so nothing is fetched from a CDN. The sanitizer allowlists HTML and attributes, scopes ids and restricts resource protocols. Choose folder resolves relative images and links from files you select; they stay on your device, and a relative URL never reaches an application endpoint.

Copy, Use as context and Save always take the Markdown source, never the flattened preview; Unedited saves retain original bytes; edits retain encoding/BOM and line endings. Preview shipped in v1.82.0 and was checked on October 11 with 30 of 30 visible browser checkpoints.

◆ FILE DROP · v1.83.0

Drop a file anywhere, and she knows what it is

One listener on every page reads each dropped file and sends it to the mechanism that fits it.

Drag a file from Windows Explorer onto the chat, the Agentic Control Panel, the Prompt Flow Panel, the welcome and login pages, the admin pages or the PDF viewer. A small overlay says what will happen on that page. The listener claims only file drags, so text you drag, such as an agent name or a palette item, is never touched, and a drop on a file input is left to the browser.

It decides by the bytes, not the name: the extension, the MIME type, magic numbers, a byte-order-mark aware text-or-binary test and a real JSON parse that recognises an MCP configuration in every common shape, UTF-16 files from PowerShell included. A bare map of names to servers counts only when every value is a server, so ordinary JSON is never hijacked. An MCP import is never silent: a confirmation lists the servers and warns that they can start programs on this computer.

Each page tells the listener what it can do, and the same registration writes the drag overlay. Up to 25 files at a time; a folder cannot be dropped because browsers do not reveal its path. Shipped in v1.83.0 and checked on October 14 with 138 router checks, 22 new tests and 58 visible browser checks.

You drop	What happens
An MCP .json	Confirm, then it joins the External MCP catalog; works on every page.
A .flw flow	Loads on the Agentic Control Panel canvas, or opens that panel in a new tab.
A .fpmt flow	Loads in the Prompt Flow Panel, asking first about unsaved work, or a new tab.
An image	Attached to your message: saved in Temp, its path written into the message.
PDF, Markdown, code, plain JSON	Opens in the chat canvas.
A folder, zip or program	Explained in plain words; nothing opens.

◆ CONFIG ▸ MODELS

Every model choice in one place

The Models dialog exposes 38 model, engine and voice settings for the core services and 21 model-backed agents, in six searchable categories.

Categories: Core 7, Vision 7, Speech 7, Workflows 6, Documents 5, Monitoring & messaging 6. Save submits all values, preserves unrelated configuration and downloads nothing. Reconnect the chat to rebuild its clients; agents pick up a choice at their next configuration load. The model list comes from the configured Ollama servers, asked by the backend with their token, so a remote server's models are offered and accepted. Since v1.76.0 Save also opens the Model Brain's Auto-tuning dialog, which tunes each configured model in front of you and names the source of every value.

Agent templates marked `"@config"` follow the global choice, while a literal value in a workflow remains an explicit override. Wrapped chat launches seed the global choices before any explicit tool argument, and the frozen build proves every model loader runs before it packages.



◆ ONE LOOK · v1.80.0

Every page shares one theme and one way to resize

Chat, both panels, login, welcome, Usage, the PDF viewer and the admin pages share one identity.

One set of design tokens defines the slate surfaces, teal accent, readable text, borders, fonts and scrollbars for every first-party page, including a themed Django admin that keeps its light, dark and automatic modes. Dialog colors follow the same tokens, and the dialog theme always loads last.

All five pane dividers share one grip and one gesture model: pointer capture, arrows for one percent, Shift for five, Home and End, and clean cancellation on Escape, blur or a lost pointer. Resizable dialogs accept all eight edges and corners and the keyboard, and a modal can no longer slide under the expanded menu at 400 percent zoom.

Shared tokens
One palette, font set and scrollbar style for every page.

Five dividers
One grip, one keyboard model, one clean cancel.

Dialogs
Eight resize directions, keyboard resizing, buttons that match.

High zoom
A modal stays above the menu at 400 percent.

Evidence
175 source tests and visible checks passed on October 11.

◆ OPERATOR DIALOGS

Everything else, without editing files

Configuration, credentials, databases and updates are handled from the browser.

Access Keys Wizard

One guided place for provider keys; blank fields keep what is set.

Contacts

Names resolve to addresses, phones and @usernames for messaging agents.

DB menu

WAL-safe backup, and a verified database staged for the next start.

External ▸ MCPs

Search the catalog, activate up to five servers, install runtimes.

ACPX-Skills

Browse, enable, diagnose and reload skills.

Check for updates

Stage the latest published release and swap it in safely.

◆ **One dismissal rule everywhere**

Escape means exactly what a dialog's ✕ means, and an outside click never dismisses anything. A downloading updater is the single dialog that refuses to close.

CHAPTER

IV

The Bestiary

87 agent types in twelve families. Each is a plain-Python program you can read, audit and change.

IN THIS CHAPTER

- 87 agents, twelve families
- Flow control
- Routing and logic
- Orchestration and intelligence
- Code, files and shell
- Documents
- Web, data and APIs
- DevOps and infrastructure
- Quality and security
- Firmware and engines
- Desktop and capture
- Voice, audio and vision
- Messaging and monitoring

◆ THE AGENTS

87 agents, twelve families

Every agent follows one skeleton: a template directory, a `config.yaml`, a session-scoped pool copy, PID and log files, and explicit source and target wiring.

68 of them also run from the chat as wrapped `chat_agent_*` tools, each in its own isolated runtime folder. Every wrapped agent appears in the Exec Report automatically, and each ships at least one example prompt in the catalog.

Display names keep their exact designed casing — STM32er, PDFer, LaTeXer, ACPXer — while directories, pools and style classes use lowercase forms.

6 Flow control	7 Routing and logic	8 Orchestration and intelligence	15 Code, files and shell
3 Documents	7 Web, data and APIs	6 DevOps and infrastructure	8 Quality and security
6 Firmware and engines	5 Desktop and capture	7 Voice, audio and vision	9 Messaging and monitoring

◆ AGENT FAMILY · 6 AGENTS

Flow control

Begin, pause, schedule and end a flow.

Starter
The entry point of every flow; launches its targets when you press Start.

Ender
Ends a flow gracefully: kills its targets, then starts backups and cleaners.

Stopper
Watches logs for a pattern and stops exactly those source agents.

Cleaner
Post-run janitor that removes the logs and PID files of finished agents.

Sleeper
Waits a configured number of milliseconds, then starts its targets.

Croner
A daily clock trigger that starts its targets at a configured HH:MM.

◆ AGENT FAMILY · 7 AGENTS

Routing and logic

Decide where the flow goes next.

Raiser
Fires its targets the moment a pattern appears in a watched log.

Forker
Automatic A/B router: whichever of two patterns appears first picks the path.

Asker
Asks you in the chat to choose Path A or Path B: a human in the loop.

Counter
A persistent tally that routes below, or at and above, a threshold.

AND
Two-input gate that fires once both sources have matched.

OR
Two-input gate that fires as soon as either source matches.

Barrier
N-input fan-in that releases once every configured source has arrived.

◆ AGENT FAMILY · 8 AGENTS

Orchestration and intelligence

Design, connect, supervise and reason.

Parametrizer
Maps one agent's structured output into the next agent's configuration.

FlowCreator
Designs a complete, validated .flw workflow from a plain objective.

FlowHypervisor
A model watchdog that reads a running flow: OK or ATTENTION NEEDED.

FlowBacker
Backs up a session's logs and configurations before clean-up.

Prompter
Sends one prompt to a model and logs a structured answer.

Summarizer
Summarizes text once, or watches logs and fires on a detected event.

ACPXer
Runs one external coding-agent CLI turn on the canvas and captures its answer.

MCP Doctor
Diagnoses an external MCP server on paper, before any live connection.

◆ AGENT FAMILY · 15 AGENTS

Code, files and shell

Find, read, change and run.

Executer
Runs a shell command, optionally in its own visible console.

Pythonxer
Runs inline Python behind a compile() and Ruff correctness gate.

Terminater
Types commands into a real CMD or PowerShell terminal the user watches, and can hand the keyboard over.

Editor
Byte-exact find-and-replace in one file; refuses an ambiguous match.

Grepper
Encoding-aware regex search, plus verbatim reads of a line range.

Globber
Finds files by glob pattern, newest first.

File-Creator
Creates a file with exact content, including a base64 channel.

Mover
Moves or copies files by glob pattern.

Deleter
Deletes by glob inside a guarded working directory.

File-Interpreter
Reads DOCX, PPTX, XLSX, PDF and more, optionally summarizing them.

File-Extractor
Extracts raw text from PDF, DOCX and similar documents.

De-Compresser
Compresses or decompresses .zip, .7z, .gz and tar archives.

J-Decompiler
Decompiles JAR and WAR archives with the bundled jd-cli.

Torrenter
Manages account-owned transfers in Tlachihuani Jobs and reports verified output paths.

Gitter
Runs git operations and custom git commands.

◆ AGENT FAMILY · 3 AGENTS

Documents

Compose, typeset and present.

PDFer
Composes PDFs from text, Markdown, HTML and images: 24 styles, 20 themes.

PPTXer
Builds editable PowerPoint decks whose text is measured, never truncated.

LaTeXer
Typesets LaTeX projects, with a self-healing eight-rung repair ladder.

◆ AGENT FAMILY · 7 AGENTS

Web, data and APIs

Search, browse, call and measure.

Googler
Resilient two-tier web search with a structured Google-dork builder.

Crawler
Fetches a page raw or as text and analyzes it with a model.

Playwrighter
Drives a real browser through scripted steps: log in, fill, click, extract.

Apirer
Calls REST APIs with any method, headers, body and timeout.

SQLer
Runs Python against Microsoft SQL Server with a ready cursor.

Mongoxer
Runs Python against MongoDB with a ready database handle.

NetSpeed-Calculator
Measures your connection with confidence intervals and bufferbloat.

◆ AGENT FAMILY · 6 AGENTS

DevOps and infrastructure

Containers, clusters, pipelines and hosts.

Dockerer
A Docker and docker-compose front-end.

Kuberneter
A kubectl front-end with a structured, truthful result.

Jenkinser
Triggers Jenkins jobs, handling the CSRF crumb.

SSHer
Runs a command on a remote host over SSH.

SCPer
Sends or receives files over SCP.

PSer
Finds a running process from a fuzzy name with a model's help.

◆ AGENT FAMILY · 8 AGENTS

Quality and security

Review, analyze, assess and protect.

Reviewer
Model-driven review of a git diff with an APPROVE or REQUEST_CHANGES verdict.

Analyzer
Deterministic static analysis: bandit, semgrep, ruff, eslint, gitleaks, pip-audit.

Kalier
Authorized Kali Linux tooling through MCP-Kali-Server.

Discoverer
The ProjectDiscovery recon suite on a private, self-installing Go toolchain.

Nmapper
A local, use-only nmap bridge; unprivileged connect scan by default.

Kyber-KeyGen
Generates CRYSTALS-Kyber post-quantum key pairs.

Kyber-Cipher
Encrypts with Kyber post-quantum cryptography.

Kyber-DeCipher
Decrypts Kyber-protected data.

◆ AGENT FAMILY · 6 AGENTS

Firmware and engines

Real boards and live editors.

STM32er
STM32 firmware from Blue Pill to H7: build, flash, observe, fail-safe.

ESP32er
ESP32 firmware through PlatformIO, from a template that compiles at once.

Arduiner
Arduino firmware through arduino-cli, with cores installed automatically.

ESPHomer
ESPHome smart-home devices from YAML, no C++ required.

Unrealer
Drives a live Unreal Engine 5 editor: 53 commands in nine categories.

Blenderer
Drives Blender over the official Blender MCP add-on socket.

◆ AGENT FAMILY · 5 AGENTS

Desktop and capture

See the screen, move the pointer, type.

Mouser
Moves and clicks in explicitly declared coordinate spaces.

Keyboarder
Types Unicode into a verified foreground window.

Windower
Focuses, tiles, resizes, lists and closes windows by title.

Shoter
Whole-desktop screenshots with published capture geometry.

Camcorder
Webcam photo or video at the camera's native resolution.

◆ AGENT FAMILY · 7 AGENTS

Voice, audio and vision

Hear, speak, play and look.

Recorder
Microphone capture to WAV at the device's native sample rate.

Whisperer
Speech to text that keeps listening until you stop talking.

Talker
Text to speech, always in Tlamatini's female voice.

AudioPlayer
Plays audio files with volume, truncate and loop control.

VideoPlayer
Plays video with its audio on a chosen display.

Image-Interpreter
Two vision models in parallel, merged by a third into one report.

Video-Analyzer
Rules on robot motion, transcribes audio tracks, or summarizes a video.

◆ AGENT FAMILY · 9 AGENTS

Messaging and monitoring

Reach people and watch systems.

Notifier
Shows an in-browser notification, optionally with a sound.

Emailer
Sends email over SMTP.

RecMailer
Watches an IMAP mailbox for new messages.

Telegrammer
Telegram as the bot or as your own account.

Whatsapper
WhatsApp through Meta's Cloud API, or explicitly from your own number.

TeleTlamatini
Brings the full Tlamatini chat, Multi-Turn included, into Telegram.

Instant Messaging Doctor
Diagnoses and safely repairs Telegram and WhatsApp readiness.

Monitor-Log
Model-powered log monitor.

Monitor Netstat
Model-powered network-port monitor.

CHAPTER

V

Deep Capabilities

The subsystems that give her reach: other agents, other servers, documents, boards, engines and senses.

IN THIS CHAPTER

- Other coding agents as her children
- A universal MCP client
- Procedures, handed over complete
- Compose, present, typeset
- A repair ladder that protects your words
- Real boards, fail-safe by design
- Unreal Engine and Blender, live
- Assessment for authorized targets
- Windows defence, operated by a human
- She listens until you stop, and answers in her voice
- Eyes that report their limits
- Precise hands on a real desktop
- Three search tiers, one time budget
- Read the requested page, report what happened
- Speed measured with its error bar
- Reaching people deliberately

◆ ACPX

Other coding agents as her children

ACPX spawns external coding-agent CLIs as out-of-process children, talks to them over stdin and stdout, keeps an NDJSON transcript of every turn, and brokers it all to the model as twelve tools.

14 peers are registered: claude, codex, cursor, gemini, qwen, copilot, pi, droid, iflow, kilocode, kimi, kiro, opencode, tlamatini. One-shot peers receive the prompt as a command-line argument and are read to completion; the Tlamatini self-host speaks strict JSON; interactive peers are drained by a transport-aware idle rule. ACPX is a Python port of OpenClaw's ACPX plugin, so its agent identifiers and skill contract are compatible.

ACPX tells the truth about its peers. A version check is not health and exit code 0 is not success: a stdlib-only classifier decides whether a child actually delivered, and a refusal is returned as `ok: false` with a named code instead of a green row. `acp_doctor(deep=True)` sends a real one-line prompt to each peer and names what is broken.

Children are never spawned through the shell, because cmd.exe silently cuts a command line at its first newline. npm and pnpm shims are rewritten to `node.exe` with the script path.

Non-delivery code	Meaning
PERMISSION_BLOCKED	The child stopped at its own permission prompt.
AUTH_FAILED · CONFIG_INVALID	It could not authenticate, or refused its own config.
NO_CREDIT · USAGE_LIMIT	No credit, or a plan or rate limit was reached.
UPSTREAM_ERROR	The model provider returned a server-side error.
NO_OUTPUT · CHILD_ERROR	Nothing legible came back, or it failed without an answer.
WORKSPACE_NOT_TRUSTED	Its working folder is untrusted, so permissions were ignored.

◆ EXTERNAL MCPS

A universal MCP client

Connect Tlamatini to any MCP server declared in a JSON catalog and use its tools at once, with no code written for the server.

The catalog uses the standard `mcpServers` shape and lives beside `config.json`, preserved across updates. Four transports connect — stdio, streamable HTTP, SSE and WebSocket — lazily on a background thread, so a slow server can never delay an answer. Up to five servers are active at once, and each remote tool appears as `ext__<server>__<tool>`.

Most servers start with `npx` or `uvx`, which a fresh machine lacks. A private runtime provisioner downloads Node and uv once from their official sources into the user's profile, verifies checksums, installs atomically, needs no administrator rights and never changes the system PATH.

Memory and Sequential Thinking ship in every installation, inactive. A default you delete is remembered and never resurrected; one you edit is never overwritten.



The adding-external-mcp skill: one guarded onboarding sequence.

◆ SKILLS

Procedures, handed over complete

29 `SKILL.md` packages are loaded at start-up and offered to the model through `list_skills` and `invoke_skill`.

Invoking an in-process skill is a planning handoff, not a hidden execution. It returns the complete procedure, the contract of outputs still owed, and an honest statement of what is enforced — argument checks, secret redaction and time budgets — and what remains declared policy, gated by the normal Configure and Ask Execs surfaces.

Skills include code review, security audit, the Kali pentest runbook, flow making, skill creation, summarization, the adding-external-mcp runbook, Roblox Studio, repository audit and refactor helpers, and integration guides for GitHub, Gmail, Slack, Jira, Notion, Todoist, Trello and weather. The ACPX-Skills menu enables, diagnoses and reloads them without a restart.

> ◆ **Credentials stay inside**
>
> An input marked sensitive, or named like a credential, is redacted from every audit event and from the returned result: no prefix, no length, no hash.

◆ THE DOCUMENT FAMILY

Compose, present, typeset

Three agents produce documents, each with its own craft, and none of them calls a file clean just because it was created.

Agent	Craft	Signature features
PDFer	Composes PDFs from text, Markdown, HTML and images	24 visual styles in five families over 20 content themes; measured tables and covers; a layout audit that reopens the file and checks overlap, off-page text, blank pages and contrast.
PPTXer	Builds editable PowerPoint decks	36 named treatments and 17 font pairings; text measured before it is placed, so long content continues onto a new slide; native PowerPoint verification when installed.
LaTeXer	Typesets real LaTeX	Eight templates and 30 signature styles; whole-project builds with bibliography and index convergence; readable diagnostics; an eight-rung repair ladder; a mostly-cut build delivered as `<name>.DEGRADED.pdf`; MiKTeX recommended.

◆ **Created is not verified**

A saved file only proves it exists. Read PDFer's `layout_clean`, PPTXer's `ground_truth`, or LaTeXer's `status` before calling a document clean.

◆ LATEXER

A repair ladder that protects your words

When a build fails, LaTeXer climbs eight rungs in a fixed order, each tried only if the one before produced no PDF.

Every repair is applied to a copy and re-linted; a repair that makes the lint worse is reverted. The author's file is never overwritten unless `repair_write_back` is set, every rung is recorded in an audit trace, and any block set aside is named with its line number.

Bisect is strictly last because it is the only rung that removes content, and it is skipped when the model rung merely could not be reached. A clean build tells the model plainly that the document is finished, so a good PDF is never "improved" into a broken one.

Since v1.76.0 a PDF is reported only when the delivered file really exists. A build that had to cut a quarter or more of its body is delivered as `<name>.DEGRADED.pdf`, never under the name you asked for, and in a document over 8,000 characters the model rung first repairs only the lines around the compiler's error with a capped request; a reply that never came forbids bisect.

1	`lint`	Static structural repair before any compiler runs.
2	`preamble`	Infer packages from the commands actually used.
3	`rules`	Deterministic rewrites of known-bad constructs.
4	`log-directed`	Compile, read the real error, fix that line.
5	`acquire`	Install a package that is genuinely missing.
6	`engine swap`	Retry with XeLaTeX or LuaLaTeX.
7	`model`	Ask a model; its answer re-enters at rung 1.
8	`bisect`	Last: set the failing block aside, and say so.

◆ FIRMWARE

Real boards, fail-safe by design

Four agents take firmware from scaffold to a running board, and each refuses rather than flash something that cannot be right.

Agent	Toolchain	Highlights
STM32er	PlatformIO ststm32 and the STM32 Template Project MCP	Blue Pill through F7, G, L, H7, U5 and WB; zero-config bootstrap; ST-LINK probe checks; newest silicon refused cleanly until its native backend exists.
ESP32er	PlatformIO Core, called directly	Copies a blink-and-print template so a new project compiles at once; serial-aware preflight.
Arduiner	arduino-cli, downloaded on demand	The board is chosen by its FQBN; the board's core is installed automatically.
ESPHomer	The esphome CLI	Smart-home devices from YAML; headless config generator; USB first flash, OTA after.

◆ **Authorized hardware only**

Flash, erase, reset and upload change a real attached device. STM32er drives mission-critical robot firmware, which is why its preflight fails safe.

◆ CREATIVE ENGINES

Unreal Engine and Blender, live

Two agents drive running editors over their own sockets, from chat or from the canvas.

Unrealer

Talks to the Unreal MCP plugin over TCP: 53 commands in nine categories — editor, Blueprint, nodes, project, UMG, system, level, asset and material.

Tlamatini's Unreal fork

XaihtUnrealEngineMCP ships the full surface plus a scaffolder for a ready-to-build UE 5.8 C++ project with a Visual Studio 2026 solution.

Blenderer

Speaks the official Blender MCP add-on's code-execution protocol on its local socket, with a rich action catalog so most tasks need no hand-written Python.

Chainable

Both return structured blocks, so Parametrizer can carry a result into the next step and a Forker can branch on it.

◆ SECURITY

Assessment for authorized targets

Offensive and defensive tooling is part of the catalog, always limited to systems you own or are explicitly authorized to test.

Kalier
Kali Linux tools through MCP-Kali-Server: nmap, gobuster, nikto, sqlmap and more.

Discoverer
subfinder, httpx, naabu, katana, nuclei and vulnx on a private Go toolchain.

Nmapper
Uses the nmap you installed; never bundles it; connect scan needs no admin.

Analyzer
bandit, semgrep, ruff, eslint, gitleaks and pip-audit, deterministically.

Reviewer
A senior-engineer review of a diff, with a verdict a flow can branch on.

Kyber
Post-quantum key generation, encryption and decryption.

◆ BLUE-HAT TOOLKIT

Windows defence, operated by a human

The `security/` toolkit is an administrator-run monitoring and response kit, not a chat tool: a person launches it, reads its evidence and decides.

Enablement makes persistent changes. Defender and the firewall keep running, but the installation is excluded from scans, allowed through Controlled Folder Access, and six attack surface reduction rules move to Audit. Those exceptions are real trade-offs, so record a baseline before elevation; there is no automatic rollback.

The monitor watches ten signal families, from Defender health and logons to persistence, ransomware indicators and administrator-group changes. Start with a detect-only baseline, investigate, and arm only when justified. A non-destructive visible harness validates the assets themselves.

Mode	Behaviour
-DetectOnly	Reports what it would block or stop; the safe first run.
default	One armed sweep; may block an attacking IP and stop known attacker tools.
-Watch	Continuous sweeps until Ctrl+C, every 60 seconds by default.
-Aggressive	Also stops dual-use tools outside Tlamatini's paths; incidents only.

◆ VOICE

She listens until you stop, and answers in her voice

Direct chat dictation, workflow Whisperer, Talker and catalog commands have distinct voice contracts.

In standalone/workflow Whisperer, no duration means the gate records while you talk and stops after 3.5 seconds of silence, with a 300-second ceiling. `record_seconds: 0` is the switch; name a duration and it is honoured exactly. The gate starts at a sensitive floor so a speaker who begins at once is heard, and if a sound driver refuses the live stream she records a fixed length and says so. Direct chat dictation instead requires the gate and refuses that fallback.

Transcription runs locally with faster-whisper, on a GPU when present and on the CPU otherwise, or through a cloud Whisper provider. Talker speaks through an Orpheus-compatible model in a female voice — tara, leah, jess, mia or zoe. A male voice is refused by design.

The catalog's first section, VOICE COMMANDS, makes the microphone the keyboard: she transcribes your instruction, reads it back, and only then acts. Anything irreversible still needs your written confirmation under those catalog instructions. Config > Mic selects automatic submission or an editable draft. Actual dispatch uses browser speech for acknowledgment, respecting Silent mode.

◆ VISION AND VIDEO

Eyes that report their limits

Image and video analysis combine independent observers, so one model's guess is never presented as fact.

Image-Interpreter

Two vision models read each image in parallel on their own connections; a third merges both. A comma-separated list reads several images.

Video-Analyzer · robotics

A deterministic motion gate plus two vision models; PASS only when both agree.

Video-Analyzer · transcription

Timestamped speech from chosen audio tracks or a web link, locally.

Video-Analyzer · summary

Speech plus sampled visual evidence: scenes, text, facts, decisions and limits; a web link is downloaded first.

Capture

Shoter, Camcorder and Recorder save to your known folders with exact names.

Paste to chat

A screenshot pasted into the chat becomes a path she can analyze.

◆ DESKTOP AND BROWSER

Precise hands on a real desktop

Desktop agents state their coordinate spaces and their evidence, because a click that lands in the wrong place is worse than no click.

Mouser
Coordinates are declared: physical screen, desktop, normalized, window or screenshot.

Shoter
Publishes the exact captured rectangle, image size and monitor geometry.

Keyboarder
Binds each keystroke to a verified window and stops on focus loss.

Delivery is not acceptance
`input_sent` proves delivery only; interrupted input is inspected before any replay.

Playwrighter
A real browser through declared steps, with a hold-open knob to watch.

Windower
Focus, move, resize, tile, pin and close windows by title.

◆ WEB SEARCH

Three search tiers, one time budget

The chat search tool and pool agent use the same Googler implementation, with a default 120-second budget and explicit outcomes.

With no engine pinned, six plain-HTTP routes race for relevant results. If none answers, real installed Chrome uses a persistent profile across eight browser routes. Open knowledge sources form the final tier: Wikipedia, arXiv, OpenAlex, Hacker News, GitHub, Internet Archive and Project Gutenberg. The log names the answering engine and tier.

A shared health ledger rests refusing engines for 3, 10 or 30 minutes; an off-topic result is rejected, and a refusing route is not asked again in the same run. Blocked, timed out, unreachable and no matches are separate outcomes. The chat wrapper supports Cancel and enforces a process deadline with a cleanup allowance.

Structured dorks retain presets, grouped filters and links_only URL delivery. An explicit engines list skips the HTTP and open-source tiers. Pin Google when Google-only operators matter; fallback engines may broaden their meaning. Licensing and authorization remain yours.

◆ WEB READING

Read the requested page, report what happened

Crawler starts with the supplied page, bounds its work and keeps refusal pages away from the model.

The default crawl_type is page, with include_seed enabled. A range crawl defaults to 25 pages and a 900-second whole-run budget. Each download has a 45-second default wall-clock limit and an 8,000,000-byte cap. These are configurable defaults; zero disables the page count or whole-crawl limit, so unlimited operation must be a deliberate choice.

Headers are read case-insensitively. Bot walls, CAPTCHA pages and access refusals are reported before analysis; timeouts and unreachable pages remain named outcomes. If nothing was analyzed, the final report says so. Crawler reads fetched HTML; it does not execute JavaScript.

Since v1.72.4, both Crawler and Googler use a stack of skipped HTML regions, handle omitted head endings and self-closing tags, and compare the extraction with a plain-text fallback. When that fallback has at least 200 characters and the parser kept less than 25 percent, the fallback takes over. Crawler warns in its log; Googler returns an extraction_note. Parser exceptions also use the fallback; healthy extractions stay unchanged.

Page first
Read the seed URL by default; range crawling is explicit.

Budgets
Defaults: 25 pages, 900 seconds overall, 45 seconds per download.

Truthful outcome
Refusals never become source content; no analysis is reported plainly.

Text safety net
Below 25% of a 200+ character fallback: recover and disclose.

◆ NETWORK MEASUREMENT

Speed measured with its error bar

NetSpeed-Calculator publishes a speed with a confidence interval and measures latency under load.

Several keyless providers run in parallel. The measurement discards the TCP slow-start ramp, samples throughput as a derivative, rejects outliers and fuses providers with a random-effects meta-analysis. It grades bufferbloat from A+ to F. A full run transfers about 100 to 200 MB, so it asks before running.

◆ MESSAGING

Reaching people deliberately

Messaging agents use official surfaces by default and make every unofficial route an explicit choice.

Telegrammer
Sends as the bot or as your own account; “send it as me” picks yours.

Whatsapper
Meta's official Cloud API by default; your own number only when you say so.

TeleTlamatini
The full chat in Telegram, on a dedicated account.

Instant Messaging Doctor
Checks tokens, contacts, templates and policy windows.

Contacts
A contact name resolves to the real address when a message is sent.

CHAPTER

VI

Promises She Keeps

Honesty, consent, safety and control, written into the code rather than into a brochure.

IN THIS CHAPTER

- A password prompt must not divert the wizard
- Truth over comfort
- Fail open or fail safe, on purpose
- Your agents, your jurisdiction
- Your data, handled carefully
- Nothing runs where nobody can see it

◆ INTENT GUARDS · v1.90.0

A password prompt must not divert the wizard

The account wizard launches createsuperuser through Executer and waits for input in its console.

The installed log showed two independent false positives. 'I will type it' referred to a password but reserved all commands for the user. A completion phrase plus the optional email field falsely created notification debt. These runtime guards overrode an initially correct Executer selection and diverted the wizard.

Typing/entering ownership now requires a command object; ambiguous input is neutral. An explicit assistant launch of a named program lifts a prior lesson. Genuine user-runs tutorials still refuse command runners and keep Terminater's Enter disabled.

A completion notification requires a positive outbound instruction tied to completion in the same sentence. Email fields, ordinary chat explanations, quoted examples, negations and the user's DONE reply do not authorize sending. Passwords stay out of chat. The catalog wizard creates an administrative superuser, not a restricted ordinary account.

Launch
Executer opens the interactive source or frozen command in a visible console.

Input
Username, optional email and password belong to the user's console interaction.

Wait
DONE / ERROR / NOWINDOW drive the next step; no automatic success claim.

Notify
Only an explicit outbound request creates completion-notification debt.

◆ HONESTY

Truth over comfort

Where a system can be confidently wrong, Tlamatini is built to say what she knows and what she does not.

Self-report first
An agent's typed result outranks a one-bit exit code.

Degraded is red
A compromised deliverable is never painted green.

Silence is named
Skipped files, dead transfers and refusals carry a reason.

Recovery is narrated
Retries are shown live and summarized in the answer.

Created is not clean
Documents report their audits before being called finished.

Delivered is not accepted
Desktop input evidence is kept distinct from success.

◆ FAILURE MODES

Fail open or fail safe, on purpose

Every guard chooses its failure direction deliberately, by asking which mistake would hurt you more.

Mechanism	Direction	Why
Binary-content guard	Open	Dropping real context is worse than embedding one odd file.
Verdict parsing	Open	A parse error falls through to the next rule; it never raises.
Port and console settings	Open	A typo must never stop the server from starting.
Runtime provisioner	Open	A failed download never blocks start-up.
Compact-mode fitter	Open	Any doubt sends the complete request, exactly as before.
Ask Execs	Safe	Any doubt resolves to Deny.
Database copies	Safe	An unclear or unchecked copy is reported as a failure.
LaTeXer bisect guard	Safe	When in doubt, protect the author's content.
Firmware preflights	Safe	Refuse rather than build or flash the wrong target.
Deleter	Safe	Refuses protected folders, repository roots and drive roots.

◆ RESPONSIBILITY

Your agents, your jurisdiction

Every agent is a plain-Python program so you can read, audit, edit, restrict or disable it. That transparency is a means of control, not a warranty of safety.

Agents have no independent authority. You decide whether, where, how and with which permissions they run. When you enable, configure, modify, chain or execute an agent, that execution is under your control and your jurisdiction.

You are responsible for reviewing code and configuration, limiting secrets and permissions, authorizing every file, target, browser, shell, API, MCP server, machine and device an agent can reach, supervising its output and complying with the law, policies, licenses and agreements that apply.

By running an agent you accept responsibility for its actions and consequences. Tlamatini's orchestration, documentation and guardrails do not authorize access to third-party systems and cannot replace your own security review, permission controls, monitoring or legal compliance.

◆ DATA SAFETY

Your data, handled carefully

Databases, secrets and scratch files each have one well-defined path.

WAL-safe database
Backups read through the write-ahead log with SQLite's online backup API and are verified with quick_check.

Staged replacement
A new database waits in DB/ToLoad; the old one is archived with its sidecars before Django starts.

Secrets
Known secret fields are redacted from exported .flw files and public builds.

Private contacts
Public builds and source snapshots carry no contact data.

Scratch files
Temporary files stay inside the application's own Temp folder.

Passwords
PDF passwords live in memory only.

◆ TESTED IN THE OPEN

Nothing runs where nobody can see it

Tlamatini's development rule is that every test, build and diagnostic runs visibly, on a real desktop, where its creator can watch each step.

Browser tests run headed in real Chrome, consoles open in the foreground, and whole-desktop photographs are taken by Tlamatini's own Shoter agent — because using the agent is the test of the agent. A stale, transient or timed-out answer is never recorded as a pass.

Source-derived guards stop drift before it ships: the status vocabulary, display names, dialog dismissal, catalog ordering, the Ask Execs allowlist, the self-modify gate and the frozen-bundle module proof are all checked by tests that read the real source.

CHAPTER

VII

Install, Configure, Run

From a fresh Windows machine to a working assistant, and from source to a signed-off release.

IN THIS CHAPTER

- Five steps to a working Tlamatini
- Every referenced Python library is accounted for
- Two modes and their service ports
- The settings that matter most
- From source tree to installer
- One inventory ships every required helper
- Packaging evidence and its limits
- Updating herself without losing you
- One number, from one source

◆ GET STARTED

Five steps to a working Tlamatini

Tlamatini itself is free. The only purchase is an Ollama plan, bought directly from Ollama.

1 Install
Run the recoverable release installer (Python 3.12.10 bundled), or clone the source.

→

2 Ollama
Install Ollama for Windows.

→

3 Pro plan
Activate Ollama Pro or higher, then run `ollama signin`.

4 Models
Pull nomic-embed-text and the cloud models Tlamatini uses.

→

5 Configure
Open Config ▸ Models and the Access Keys Wizard, then tick Multi-Turn.

```
git clone https://github.com/XAIHT/Tlamatini.git
cd Tlamatini
python -m venv venv && venv\Scripts\activate
pip install -r requirements.txt
python Tlamatini/manage.py migrate
python Tlamatini/manage.py runserver --noreload
# open http://127.0.0.1:8000/   (default login: user / changeme)
```

◆ DEPENDENCY COVERAGE

Every referenced Python library is accounted for

The October 18 dependency audit covers all 85 agents, build helpers, optional imports and tests.

The refreshed static guard scanned 855 Python files and 72 directly referenced distributions. The complete requirements file pins all 263 distributions, including CPU PyTorch. A clean Python installation, dependency consistency check and runtime imports passed.

ESPHome has a separate requirements-esphome.txt manifest because its py7zr and PlatformIO pins conflict with the main environment. Keep it in its private runtime. Blender bpy, Unreal unreal, device toolchains, npm packages and model weights follow their own runtime contracts; they are not missing pip packages.

Coverage guard
scripts/check_requirements_coverage.py checks source and build inventories.

Verification
October 18: no missing declarations, syntax errors or unreviewed dynamic imports.

Earlier evidence
September 27: main/ESPHome dry runs and temporary-environment dependency smoke checks passed.

Limits
A clean Python dependency installation does not certify the compiled installer or every external integration.

◆ RUNTIME

Two modes and their service ports

Tlamatini runs either from source or as a frozen executable built by PyInstaller. Only path resolution differs, and `CONFIG_PATH` overrides the configuration location in both.

An installed build resolves `config.json` beside `Tlamatini.exe`; a source run uses `Tlamatini/agent/config.json`. A second, independent axis is self-modification: a build made with `--self-modify` carries her complete, rebuildable source beside the executable, and her self-knowledge travels with it. Without the flag, neither ships. A source run is always self-able, and a toolbar Self-modify switch decides per request whether her self-knowledge is sent; it locks off when the model cannot hold it.

Direct dictation additionally uses a private ephemeral loopback listener for its worker handshake. It closes after authentication; it is not a fourth public service or fixed port.

Both MCP services honor their configured hosts and ports in source and frozen mode; their client URIs must match. Files-Search also honors its worker count and bounds RPC waits. UTF-8 BOM configuration files and CONFIG_PATH overrides are supported by both clients.

Port	Protocol	Service
8000 (default)	HTTP + WebSocket	Web interface and chat; change it with django_port.
8765 (default)	WebSocket	System-Metrics MCP; match server port and client URI.
50051 (default)	gRPC	Files-Search MCP; match server port and client URI.

◆ CONFIGURATION

The settings that matter most

A handful of `config.json` keys shape her behaviour; everything else has sensible defaults.

Key	Default	Purpose
django_port	8000	Web port for every launch path; fails open to 8000.
unified_agent_max_iterations	4096	Upper bound on Multi-Turn tool-loop turns.
unified_agent_llm_step_timeout_seconds	900	Watchdog for one model attempt.
llm_client_timeout_seconds	600	How long one model call may take.
model_brain	auto	Tunes each model from formal sources; off restores the legacy sampler.
ollama_repeat_penalty	1.2	Legacy sampler, sent only when model_brain is off.
ollama_num_ctx	1048576	Legacy requested window; the brain clamps it to the real one.
context_sidecar_llm_timeout_seconds	60	Time limit on the context helpers' own model calls.
context_compact_mode	auto	Compact mode locks on for models that cannot hold everything.
binary_context_detection	true	Screens files by content before embedding.
console_quick_edit	false	Keeps a click from pausing a frozen console.
console_colors	true	Colours console lines by level; the log stays plain text.
runtime_autoprovision	true	Lets MCP servers provision Node or uv privately.

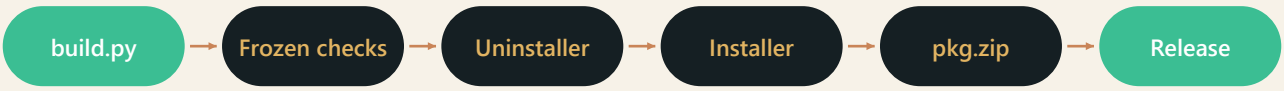
◆ **Measured, not chosen by taste**

The legacy sampler comes from repeated trials on real workloads, and one fast run never justifies a change. The Model Brain now tunes each model from formal sources.

◆ BUILD AND RELEASE

From source tree to installer

Three build scripts produce a standalone Windows release, and the build proves what it ships before it packages anything.



Every stage verifies the one before it.

PyInstaller bundle

Collected static files, templates, agents, skills and helpers, with numpy and OpenCV embedded in both shipped Pythons.

Carriage proof

The build opens the archive it produced and aborts if a required fail-open module is missing.

Runtime gates

Every agent runtime and every model loader is executed inside the new frozen executable, and the Prompt Flow Panel check runs there too.

Integrity receipt

A SHA-256 receipt covers every payload file and is checked again when the installer is assembled.

Size ceiling

The final release ZIP stays under 1,990,000,000 bytes; required files are never dropped to fit.

◆ BUILD REPAIR · SEPTEMBER 27

One inventory ships every required helper

A September 27 repair, carried since v1.72.1, ships the Whisperer settings helper without weakening package verification.

A build correctly refused a runtime receipt because chat_voice_settings.py was missing. The verifier required the loose helper, but build.py used a separate copy dictionary that omitted it. Compiling agent.chat_voice_settings does not satisfy the standalone Whisperer worker running under the carried Python.

build.py now calls copy_root_sources using the verifier's ROOT_SOURCES inventory. All inputs are checked before copying. Every mapped destination is mandatory in both ZIP and extracted-stage receipts, so removing a required file and its receipt entry still fails. The pre-freeze hash baseline continues to reject source changes during a build.

One carrier
Copy and verification share ROOT_SOURCES; a new required helper needs no second list.

Worker import
Carry chat_voice_settings.py at the installation root as well as compiling the Django module.

Integrity
Reject missing, edited or unmanifested files, including otherwise consistent incomplete receipts.

Rebuild inputs
Mirrored self-update/self-modify sweeps include the shared inventory and report omitted Git checks.

◆ RECORDED VERIFICATION

Packaging evidence and its limits

The September 27 file-level checks establish carriage and integrity; a full installer is a separate verification.

The recorded focused suite ran 146 tests: 145 passed and one Git-dependent census was skipped. It exercised the real carrier with source files, imported the worker from a staged installation, and verified ZIP creation, extraction and receipts with synthetic runtime binaries. Missing or corrupted helpers and extra files were rejected.

The recorded self-update sweep with --no-git had no findings. A self-modify snapshot copied 1,579 files with zero copy errors and accounted for all 778 runtime source inputs. These are that audit's dated counts. Neither those checks nor this dossier regeneration ran a complete PyInstaller build, installed a release or performed a live update swap.

Focused suite
146 tests recorded: 145 passed, one skipped; file-only harness exited successfully.

Real files
Every mapped source destination matched; the staged worker resolved its settings helper.

Snapshot
Recorded audit: 1,579 files, 11 redacted, 778 runtime inputs covered, zero copy errors.

Evidence
docs/build-root-assets-verification.md records scope, skipped checks and local transcripts.

◆ SELF-UPDATE

Updating herself without losing you

About ▸ Check for updates fetches the latest published release, stages it, and hands the locked-file swap to an external script.

The swap preserves `config.json`, the database, contacts, the external-MCP catalog, Temp, Templates, generated content and the separately built uninstaller. The database is copied through SQLite's online backup API before shutdown, then new migrations are applied to it on the next launch, so chat history and settings survive while new agents and prompts arrive.

Downloads and staging receipts are verified before shutdown, and a failed prerequisite stops before any file is replaced. While the download runs, the update dialog is sealed and refuses to close.

◆ VERSIONING

One number, from one source

Tlamatini uses one Semantic Versioning number for the latest released version.

The runtime resolver, the About window, the `/agent/version/` endpoint and every build artefact report the same base tag. The reported version never carries a development or dirty suffix, and `TLAMATINI_VERSION` selects the release number.

This dossier presents 1.92.4 as the latest released version throughout, including its covers, feature chapters and repository inventory.

CHAPTER

VIII

Current Release

Latest released version 1.92.4 — downloads, editing, updates and Windows execution.

IN THIS CHAPTER

- Downloads, typing and Windows delivery
- Prompt Flow fixes and short tool names
- Real settings for editor and terminal behavior
- Edit, teach and preserve the learner's control
- How the line arrived here
- One released version: v1.92.4

◆ RELEASED · v1.92.4

Matching versions before the build starts

One predicted application version for English and Spanish, with explicit release identity.

BuildTlamatinis.cmd calls each checkout's own resolver and previews its version emitter in memory. It compares the actual number the executable will report; a tag on HEAD is not required. The common value is pinned with --version before either complete wrapper starts.

Private or Public mode runs both editions in parallel with --self-modify. New ZIPs are CRC-checked, SHA-256 hashed and moved to edition/type/start-minute folders. Each successful edition keeps its archive even if the other fails; the overall command then reports failure.

Complete wrappers embed release_type in the generated version module. The application log begins with version and Private/Public banners. Source and unspecified builds are labeled explicitly; folder names, environment variables and credentials do not guess the type.

◆ RELEASED · v1.92.4

Recover safely and open readable Markdown

Build dependency isolation and lossless text loading are part of the release contract.

Standalone builds use application-owned Temp/build-env; the combined launcher supplies separate caches under the release root. If dependency setup fails and no healthy isolated environment exists, the build stops before the machine-Python NumPy purge or pip --user path. A previously healthy isolated cache remains reusable.

Readable README.md files containing occasional control characters remain editable. Strict BOM-aware decoding and byte-preserving saves stay intact. A decoded NUL, control-dominated payload or malformed encoding still causes a refusal before the canvas is replaced.

Dated evidence: the November 8 complete-build campaign passed all four real builds and packaged-executable checks at version 1.92.3, with correct release types. Integrity checks, public-data audits, 26 build-isolation tests and focused lint passed. This document refresh does not relabel those archives or claim a new install, reboot or live-model acceptance run.

◆ CARRIED FROM v1.92.3

Downloads, typing and Windows delivery

Downloads, typing, updates and Windows execution

Torrenter adds account-owned downloads to chat and the Agentic Control Panel. Live Editor and Terminater accept up to 250 WPM. The updater carries the uninstaller and preserves the vault. Release builds lease the Python environment for their requirements fingerprint.

Builds with different requirement fingerprints coexist. An OS lock protects preparation and the full build. The 1.92.3 update package includes the verified uninstaller and preserves existing data.keys.

◆ OCTOBER 2026 · v1.91.1

Prompt Flow fixes and short tool names

Tag 9915938a, dated October 27, 2026; published as the Latest release.

A patch on v1.91.0. Every chain loads the history of its own conversation, so a Prompt Flow run keeps its own memory and is never answered with another conversation.

An equals Decision accepts an answer that opens with the expected word, and the Prompt before it asks for that word alone on the first line. A flow run's usage is recorded for the account that owns it.

Telegrammer sends as the bot when the request says 'using the bot'. A wrapped agent called by its short name is resolved and logged as [TOOL-ALIAS]. No agent, tool, skill or migration was added.

History
Each run and each user reads only its own conversation.

Decisions
'## YES - ...' equals YES; 'Yesterday' does not.

Routing
'Using the bot' sends as the bot; short tool names still run.

Inventory
87 agents; 111 tools; 221 migrations; 64 JS modules; 29 stylesheets.

◆ CARRIED FROM v1.91.0

Real settings for editor and terminal behavior

Tag a771d27c, dated October 23, 2026, and published on GitHub that day.

The Live Editor, Terminater and Voice tabs share one schema, authenticated account storage and optimistic revisions. Migration 0220 adds InteractionSettings. Account behavior reaches the actual editor and the one TerminalHub; browser/account appearance stays local.

October 23 source evidence: 17 visible browser checks passed, including genuine OS speech finishing before the first terminal key. The implementation regression run passed 929 of 931 tests; two existing README documentation assertions failed. This documentation sweep restores those sections; its 94 focused tests passed, including both README assertions.

v1.91.0 was published on GitHub on October 23, 2026. These regenerated documents do not claim a new executable build or installed acceptance. The existing tag is preserved.

Dialog
Pacing, teaching, terminal behavior, narration, appearance and draft previews.

Runtime
Real settings snapshots, revision checks and cancellable narration receipts.

Compatibility
Legacy explicit flows keep their values; new flows can inherit defaults.

Inventory
87 agents; 111 tools; 221 migrations; 64 JS modules; 29 stylesheets.

◆ CARRIED FROM v1.90.0

Edit, teach and preserve the learner's control

v1.89.0 introduced editing and native saving; v1.90.0 carries Interactor and the intent corrections.

The annotated v1.90.0 tag resolves to ebc5c3b4; this documentation refresh began at d274f89c, the follow-up commit also present on origin/main. A tag and source tests do not establish that a separately installed executable has been rebuilt or accepted.

Interactor adds one module tool, not a workflow-agent template. Migration 0218 seeds its Tool row and catalog example. Full enabled-tool binding, Compact-mode selection and cost labels share tool_gate_table. The offline editor bundle includes the tutoring extension.

Editing
Native Save As, exact source, undo history and conflict-aware persistence.

Tutoring
Fresh unsaved reads, small typing segments, learner fields and local suggestions.

Recovery
Bounded tab-visibility pauses and usable narrow-canvas layout.

Inventory
87 agents; 68 wrapped; 111 built-in tools; 221 migrations. Counts describe the inspected source.

◆ DATED VERIFICATION

Keep source, model and frozen evidence separate

These are recorded runs from October 21–22, not tests silently rerun by documentation generation.

Scope	Recorded result and limit
Canvas editing / Oct 21	29 distinct visible checkpoints; native saving, file types, encoding, conflicts and PDF copies.
Interactor / Oct 21	35/35 checks including three configured real-model cases; later UI runs add interruption and recovery coverage.
Final DEV modes / Oct 21	19/19 visible checks; 136 passed + 2 skipped among 138 focused Django tests.
Older frozen v1.89.0	Real frozen-loader run: 16 passed, 2 failed after visual review (tab absence and narrow layout).
Wizard guards / Oct 22	179 tests passed; exact installed-request replay passed. Controlled tool effects; no real account or notification.
New v1.90.0 package	No new installed-package acceptance is claimed by this documentation pass.

◆ CARRIED FROM v1.88.0

Terminals you can watch, and a private release that carries your data

Tlamatini v1.88.0 introduced Terminater; its tag was created on October 20.

Terminater is workflow agent number 87. It raises real CMD and PowerShell terminals in a new Terminals area (Views → Terminals, or Ctrl+`) on the chat, Agentic Control Panel and Prompt Flow Panel pages. Tlamatini types every key like a person, about 50 words per minute and never pasted, waits for the command, reads its whole output and reports the exit code. Esc takes the keyboard; a command can hand it to you on purpose with <<USER:hint>>.

A command that stops answering is flagged hung and can be aborted without losing the terminal; one that exits with an error is a red Exec Report row even though the terminal worked. One hub in the Django process owns the shells, so closing a page never closes one. It is wrapped chat agent number 68, the eighth Prompt Flow Panel operation, and has three safe cards in the Catalog of Prompts.

The build now proves it inside the frozen app: check_terminater drives a real PowerShell, a real cmd.exe, the shipped pool agent under the carried Python and a WebSocket round trip, and packaging stops if it fails. On October 19, 774 unit tests passed, and 1,668 passed after the documentation pass. The saved chat-page and canvas visible suites were not re-run as one clean pass after their last fixes.

The v1.88.0 tag was created on October 20 and points to the commit before this work; the work landed on main in the next commit, so a build of main then reported 1.88.0 and carried all of it.

The private (keyed) release now carries the data of the machine that builds it: every External MCP server of the installed copy and its active choice, merged into a gitignored private catalog, and the whole data.keys vault. The public builder refuses both. As of v1.92.2, updates preserve an existing vault.

Terminals
Real CMD and PowerShell you can watch, typed like a person, with a hand-over.

Current counts
87 agents · 68 wrapped · 111 tools · 221 migrations.

Gate
check_terminater runs inside the frozen app; packaging stops on failure.

Private release
External MCP servers and data.keys travel in the keyed build only.

◆ CARRIED FROM v1.87.0

A leaner catalog, a safer removal, a smaller release

Tagged on October 18: the catalog shrank to 85 agents, and removal became fast and recoverable.

That release held 85 workflow agents: Gatewayer, Gateway-Relayer, Node-Manager and Zavuerer were removed. Migration 0214 deletes their rows and catalog prompts without renumbering anything that remains.

The Uninstaller moves the installation aside on the same drive instead of copying it, and offers Retry cleanup when a locked file stays behind. The frozen release no longer carries the media and browser libraries twice. About → Usage waits for real data, and every tracked Python file has a quality gate.

The v1.87.0 tag was created one minute before the work it names. Both follow-up commits landed on main right after it, so a build of main then reported 1.87.0 and carried all of it.

85 agents
Four retired agents removed in a forward migration.

Removal
Move aside, verify, Retry cleanup for locked leftovers.

Release size
Media and browser libraries ship once, in the carried Python.

Quality
npm run lint:python checks every tracked Python file.

◆ NEW IN v1.87.0

Uninstalling is fast and recoverable

The Uninstaller moves entries into a recovery folder instead of copying and hashing them.

Each move is recorded in a schema 2 journal before and after it happens. A destination collision is refused, and the journal stays until cleanup succeeds. If a reversible step fails, the entries and the Windows state are put back.

If the application is removed but a locked file remains, the window says Application removed; cleanup incomplete, lists what remains and offers Retry cleanup. Reopening the Uninstaller can finish the job later. agents/ and the six content folders that hold files are kept.

A filesystem benchmark removed 64,264 package files (3.88 GB) in 22.3 seconds with all 254 preserved hashes unchanged. It measures the filesystem phase; a compiled-client run remains the acceptance test.

Journal
Every move recorded before and after.

Retry
Locked leftovers can be finished later.

Kept
agents/ and the content folders that hold files.

Benchmark
64,264 files in 22.3 s, October 18.

◆ NEW IN v1.87.0

One owner for every library

The frozen web process stops carrying libraries that only the pool agents use.

cv2, ffpyplayer, av, playwright, faster_whisper, ctranslate2 and onnxruntime run under the carried Python, so build.py keeps them out of the frozen interpreter. Chromium and FFmpeg revisions come from the carried Playwright manifest.

Only bytecode caches whose source sits beside them are removed, Git's copied uninstaller files are dropped and the package is written with DEFLATE level 9. A release-payload test pins who owns each library.

About → Usage shows Checking usage… until real report data arrives, explains a failed connection under Connection details and retries by itself. Its account request is a bodyless POST /api/me: a {} body made the local server reset the connection.

Ownership
Pool-only libraries live in the carried Python.

Package
Bytecode, uninstaller copies and compression tuned.

Usage
Checking usage… until real data arrives.

Account
A bodyless POST /api/me.

◆ AFTER THE v1.87.0 TAG

The consistency sweep

A source-wide sweep re-derived every count and fixed what it found.

Grepper's glob and Globber's pattern now understand brace sets such as *.{js,py}, lists such as *.py,*.js and folder globs such as src/**/*.py. Before, each searched zero files and answered no_matches. Globber walks the tree once for several patterns: three minutes became ten seconds.

Six wrapped tools whose label names an action (Send Email, Move File, Summarize Text and the three Kyber tools) now obey their agent's Configure Agents checkbox. A loaded flow keeps FlowCreator's label, and the Python quality gate runs in Windows Terminal.

The daily chat test reads this repository again instead of an old copy, and its question banks cover all 85 agents and all 66 wrapped tools. Configure Agents and Configure MCPs now open with the boxes the server saved, so a quick Continue can no longer switch most agents off.

Patterns
Brace sets, lists and folder globs in Grepper and Globber.

Gates
Six wrapped tools follow their agent's checkbox; Configure opens with the saved ticks.

Harness
The daily chat test reads this repository.

Quality gate
Runs in Windows Terminal, the Windows 11 default.

◆ CARRIED FROM v1.86.0

Mover folders and the Architecture atlas

Tagged on October 17: a missing destination becomes a folder, and the atlas rebuilds itself.

Mover creates a destination that does not exist yet, such as scripts/architecture_atlas, as a folder, and the file keeps its own name. A rename still happens when the destination names a file: it exists as a file, has an extension, or the source has none either. A trailing slash always means a folder.

python scripts/architecture_atlas/build_architecture_pdf.py regenerates Tlamatini-Architecture.pdf with Tlamatini's own LaTeXer and verifies that every tracked file appears in it. Exit codes 0, 1, 2 and 3 mean ok, typeset failed, a file left unplaced and a file missing.

Mover
A new destination name becomes a folder.

Rename
Only when the destination names a file.

Atlas
Rebuilt by LaTeXer from the repository.

Proof
Every tracked file appears in the atlas.

◆ CARRIED FROM v1.85.0

Refresh, Views and a centered avatar

Tagged and published on October 16, with a more flexible chat workspace.

Refresh rereads the open document. Views makes room for full-width chat while keeping the canvas alive. Reset restores the layout without clearing content. The portrait and its frame remain centered over Mic through Send as the message box shrinks.

The release retains the page-wide file-drop router, setup recovery, Markdown rendering, Usage dashboard, Model Brain, Compact mode, voice dictation and both flow editors.

Refresh
The original document, read again without a chooser.

Views
Show or hide the canvas and keep its state.

Reset layout
Restore both panes and dock the avatar.

Portrait
Center the whole frame above the complete action row.

◆ CARRIED FROM v1.85.0

Refresh the original document

Open selects a file. Refresh reads that same source again, without writing to it.

Native-picker access retains a browser-granted handle. Drops retain their handle or entry during the original event, including the PDF-frame relay. A snapshot-only source explains why Refresh is unavailable. A generated document rereads its authenticated saved-content endpoint.

Text and Markdown are compared in full, including empty files. Equal size or timestamp is never proof of equal content. An unchanged document stays in place without rebuilding context. Changed content replaces the complete document only when preparation succeeds.

Busy, unchanged, updated, permission and error states have visible feedback. Cancel, Clear, opening another document and a new operation invalidate pending reads. Failure preserves the previous document and explains the next action.

Open
Choose a document and retain read access.

Refresh
Reread the retained source without another picker.

Failure
Keep the previous content and allow retry.

Context
Prepare changed content before replacing active context.

◆ CARRIED FROM v1.85.0

Keep the reading position

Refreshing Markdown or PDF preserves the view while the underlying document changes.

Markdown retains Preview or Source, reading position and explicitly selected resources. Its source bytes remain the basis for Save. Folder-selected resource files remain snapshots and can be selected again when those separate files change.

PDF Refresh stages a new range-backed PDF.js viewer before retiring the old one. It keeps page, zoom, rotation and scroll mode, clamping the page to a shorter document. A corrupt PDF or cancelled password entry leaves the prior viewer visible. The staged iframe stays attached during the swap so the browser does not reload it.

PDFs always reload on Refresh. No whole-file hash, eager whole-document read or metadata-only equality claim is introduced. Active-context replacement keeps the established text/image options.

Markdown
Preview/Source, position and selected resources stay.

PDF
Page, zoom, rotation and scroll mode stay.

Recovery
A failed replacement retains the previous viewer.

Evidence
20 visible Chrome checks on October 15.

◆ CARRIED FROM v1.85.0

Choose a full-width chat

Views → Canvas shows or hides the canvas without unloading it.

Hiding the canvas gives the chat the full width. The document, PDF viewer, Markdown mode, Refresh source, draft, history and loaded context remain intact. Open, a supported document drop and a generated-file link reveal the canvas when their result needs it.

The choice is stored per account in that browser and synchronized across its tabs. It survives reload and browser restart. A different browser profile or origin has separate preferences.

Changing visibility cancels divider gestures. Focus moves to the Views menu only when its current region disappears, and the check mark exposes its state to assistive technology.

Full width
Hide the canvas to give the chat more room.

Continuity
Keep the open document and active context.

Preference
Saved per account and browser origin.

Navigation
Opening a document reveals its canvas.

◆ CARRIED FROM v1.85.0

A centered portrait at every height

The portrait and its frame share the center of the entire Mic-to-Send action row.

Previously, reducing the composer height shrank the portrait but left it anchored to the right edge. The layout now measures from Mic's left edge to Send's right edge and centers the whole dock above that span. Both dividers, narrow layouts and zoom follow the same measurement.

The portrait remains square with a 120-pixel minimum. Its floating size remains independent. Hiding the canvas docks the portrait and disables floating resize through pointer, keyboard and API paths. Showing the canvas enables resizing again and leaves the portrait docked.

Views → Reset layout shows both panes, restores their initial sizes and docks the avatar. It does not clear the document, draft, history or context.

Alignment
Mic's left edge through Send's right edge.

Sizing
A square portrait, never below 120 pixels.

Reset
Restore layout while keeping content.

Evidence
19 visible checks; center error below 0.02 CSS px, October 16.

◆ CARRIED FROM v1.85.0

The complete feature travels together

Required runtime and self-modify inventories cover the Views template and scripts.

The runtime receipt requires agent_page.html plus agent_page_layout.js, avatar_size.js, agent_page_canvas.js and agent_page_init.js. The snapshot also requires the two visible Views harnesses and the identity instructions. Refresh retains the existing PDF and Markdown carriers.

There is no new migration or Python dependency. Its cache suffix was -avatar-center-1 and the Views template suffix is _views1. Visibility is browser-local and does not depend on Self-modify.

Dated evidence includes 160 packaging/self-update/self-modify tests and 13 visible frozen frontend checks on October 16. The latter used an actual executable with byte-verified current frontend assets in a disposable fixture. The carriage guide records the exact scope.

Runtime
Required template and four cooperating scripts.

Snapshot
Sources, identity and both Views harnesses.

Database
v1.85.0 introduced no migration.

Version
v1.85.0 across the application at the time.

◆ CARRIED FROM v1.84.0

Catalog and fixture maintenance

The October 14 maintenance commit updated catalog/configuration data and scrubbed test fixtures.

Commit 1a8c5093 changed FlowCreator catalog data, the local External MCP catalog and test fixtures. It introduced no new product subsystem, agent type, migration or dependency. Private catalog values are excluded from this dossier.

The current audit reads the actual implementation and keeps historical test results dated. A fixture value is test data, not a deployed credential or evidence that a provider ran.

Maintenance
Catalog data and test fixtures changed in commit 1a8c5093.

Product surface
No new subsystem, agent type, migration or dependency.

Privacy
Public documents exclude private catalog values.

◆ CARRIED FROM v1.83.0

One smart listener for every file you drop

Tagged and published on October 14: a dropped file never errors, never opens as text by mistake and never navigates away.

Dropping an MCP .json on the chat used to report an error, on the canvas it opened as plain text and killed the import, and on every other page the browser left for the file. One listener now sits in the head of the chat, both panels, the welcome and login pages, the admin pages and the PDF viewer frame, as the File Drop section describes. It claims only file drags.

It decides by the bytes, not the name, and sends each kind to its own mechanism: an MCP .json to a confirmation and the External MCP catalog, a .flw or .fpmt flow to its editor, an image to the message, a PDF, Markdown or code file to the chat canvas, and anything else to a plain explanation.

It replaced four per-page handlers. Checked with 138 router checks against the real script, 22 new Django tests and a visible Chrome run of 58 checks with 24 Shoter photographs, all on October 14.

One listener
Capture phase, file drags only; text drags are untouched.

By the bytes
Extension, MIME, magic numbers, BOM-aware text test, real JSON parse.

MCP import
Never silent: a confirmation lists the servers first.

Everywhere
Chat, both panels, welcome, login, admin and the PDF frame.

Evidence
138 router checks, 22 tests, 58 visible checks, 24 photographs.

Tag
v1.83.0 at a2914019, pushed and published October 14.

◆ CARRIED FROM v1.83.0

The portrait shrinks with the message box

Commit 63a62cc1 on October 13, credited to Tlamatini's-AutoBot; shipped in v1.83.0.

The divider above the message box used to reserve the portrait's current width as the box's minimum height, so a wide portrait kept the box from shrinking. It now reserves only a 120-pixel portrait plus its controls, and the docked portrait follows the divider: its side is the smaller of a quarter of the form width and the height above the Send and Mic row, never below 120 pixels.

At the initial 1440-pixel window the portrait went from 282 to 120 pixels, returning about 162 pixels to the chat history. Nineteen docked-avatar checkpoints and all eighteen existing composer checkpoints passed in a visible Chrome window.

Down
Dragging the divider down shrinks the portrait to a 120-pixel square.

Up
Dragging it up gives the portrait back, up to its original column.

Keyboard
Arrows, Shift+arrows, Home, End, double-click and Escape still work.

Evidence
19 docked-avatar and 18 composer checkpoints passed.

◆ CARRIED FROM v1.83.0

A wrongly worded request is not fatal

Commit 91ff8453 on October 12, credited to Tlamatini's-AutoBot; shipped in v1.83.0.

When the launcher refuses an Image-Interpreter or Video-Analyzer call before the agent starts, because a setting does not exist or a mandatory one is missing, the refusal is marked as a request error. It no longer raises the red fatal-analysis dialog and its sound; the model simply corrects the call. A real start failure still raises the dialog.

Video-Analyzer's description now says it has no transcription engine setting, which belongs to Whisperer. Five new tests pin the behavior, and it reaches installations with v1.83.0.

Request errors
Refused before any model ran; the model retries.

Real failures
Still raise the accumulated fatal dialog.

No engine setting
engine= belongs to Whisperer, not Video-Analyzer.

Tests
Five tests in RequestRefusalIsNotFatalTests.

◆ CARRIED FROM v1.82.0

A way back for installing and removing

Tagged and published on October 12: new setup windows, and Markdown that opens as a page.

The Installer and Uninstaller share one window that prepares and verifies a recovery copy before changing any file, restores an interrupted run, names a locked file and keeps every error on screen with Retry, as the Setup Windows section describes. Markdown files now open in the chat canvas as a rendered page with their source kept intact.

The release was published on GitHub on October 12 and was the latest until v1.83.0 replaced it on October 14. Its zip was a self-modify build whose package carried the setup windows, the Markdown canvas, the shared theme of v1.80.0 and the web links of v1.79.0. There is no v1.81.0 tag.

The setup campaign ran visibly on October 11 and 12: 52 interface checkpoints, 28 recovery checks, 21 lifecycle checks and 61 mechanics tests passed, with five real installations and four real removals passing their independent audits. The Markdown canvas passed 30 of 30 visible checkpoints. The campaign did not install the published zip.

Setup windows
One recoverable window for installing and removing.

Markdown
Rendered Preview, original Source, offline sanitizer.

Published
October 12; the latest release until v1.83.0.

No v1.81.0
The number was skipped; v1.80.0 and v1.79.0 come before it.

Campaign
Five installations and four removals passed; one removal stopped safely.

Followed by
v1.83.0 on October 14 carries the three later changes.

◆ CARRIED FROM v1.80.0

One look, and one way to drag

Tagged on October 11 without a release of its own; it reaches users through v1.82.0 and v1.83.0.

Every first-party page now draws from one set of design tokens, and the five pane dividers share one grip and one gesture model, as the One Look section describes. The side divider was first made to match the composer divider, then the whole site followed the same evening.

The source campaign passed 175 tests with two platform skips, 17 composer checkpoints, 38 zoom and responsive checkpoints at 25 to 400 percent, 13 commentary checkpoints and all eight dialog resize directions, in a visible Chrome window. It certifies the source pages, not a model answer or an installed build.

Tag
v1.80.0 at 82099963, pushed; no published release.

Tokens
Slate surfaces, teal accent, one font set, one scrollbar.

Dividers
Five seams, one grip, arrows and Shift, Home and End.

Dialogs
Eight-direction and keyboard resizing; matching buttons.

Evidence
175 tests and visible zoom, composer and dialog checks.

Delivered by
The published v1.82.0 and v1.83.0 releases.

◆ CARRIED FROM v1.79.0

Video-Analyzer takes a web link

Tagged on October 11 without a release of its own; it reaches users through v1.82.0 and v1.83.0.

Given a YouTube or other web link, Video-Analyzer downloads one item itself: only the sound for a transcription, the picture for a summary or a robotics check. It keeps a private, self-updating downloader, updates it once and retries when a download fails, joins picture and sound without ffmpeg, and names every failure instead of returning silence.

The chat parser no longer mistakes a link's ?v= for a setting, a request written in plain words fills in the link and the mode, and the result adds the source address and title. The model is told to hand links to Video-Analyzer and never to install a downloader itself. 141 unit tests and a visible 12-of-12 chat run passed on October 11.

One item
Sound for transcription, picture for summary or robotics.

Self-updating
A private downloader, refreshed once on failure.

Named failures
Unavailable, login required, network, timeout and more.

Prose works
Plain-language requests fill in the link and the mode.

Evidence
141 unit tests and a visible 12-of-12 chat run.

Delivered by
The published v1.82.0 and v1.83.0 releases.

◆ CARRIED FROM v1.78.0

Usage: credits and activity, while she works

Tagged on October 10; the Usage commit lands one commit after the tag. The published v1.78.0 zip carries it.

The first October 10 campaign passed 18 Usage tests, 202 existing regressions and 24 visible browser checks, plus source/collected-static byte parity. The live Pro account was compared with Ollama Settings before and after a real model request, and Usage opened and refreshed while that request ran. Other-plan, failure and account-switch cases used explicitly synthetic fixtures, not live Free, Max or Unlimited subscriptions.

A follow-up the same day made Usage recover when Ollama starts late. Temporary connection, timeout and invalid-response failures now retry after 15 seconds instead of a full minute, an upstream Retry-After is respected even when Refresh is clicked repeatedly, and a refused connection is checked again at once on Refresh or reopen. It passed 27 Usage tests and 12 visible browser checks: Refresh recovered in 0.47 seconds, reopening in 0.52 seconds and the automatic retry in 15.02 seconds.

UsageDaily, migration 0213, records counters by user/model/UTC day through a bounded worker queue; it is best-effort telemetry, not a billing ledger. Both inclusion sweeps passed. Those campaigns ran from source. This refresh checked that the published zip matches the local build byte for byte (SHA-256); it did not install the release or rerun the browser tests.

220 tests
18 Usage plus 202 existing regressions passed on October 10.

24 browser checks
Visible live Pro comparisons plus clearly synthetic edge cases.

Recovery
27 Usage tests and 12 visible checks; outages retried after 15 s.

During a request
Usage opened/refreshed while a real model answer completed.

In the release
The published zip's files match the latest October 10 commit.

Privacy
Account observations and authenticated screenshots stay out of this dossier.

◆ CARRIED FROM v1.78.0

Undock her, and the message box shrinks to fit

Two commits after the v1.78.0 tag, and carried by the published zip: the input area resizes once she floats.

With the avatar undocked, the input form can now be dragged down to its own action buttons plus a six-pixel guard; the old 30% floor is gone. The toolbar, context gauge, attachment and dictation rows count with their measured heights. Docking her again restores room for the portrait. The divider answers the pointer, the arrow keys, Shift+arrows, Home, End and double-click, and reports its height to screen readers.

Short windows scroll the composer inside itself and keep the chat history. The page occupies the real viewport: collapsed navigation opens over the workspace instead of pushing it down, and a 640-pixel minimum workspace keeps both panes on narrow screens. jQuery UI dialogs stay inside the window after opening, dragging or resizing, short native dialogs scroll as complete cards, and the avatar never covers a modal dialog.

In a verified foreground Chrome window, 18 composer checks and 38 viewport checks passed at 100%, native 400% and 25% zoom and in a phone-sized window, with eight dialogs at each size. That is a desktop browser at phone-sized dimensions, not a physical phone; 20% zoom was unavailable and is not claimed. These layout checks do not certify model inference or audio capture.

Shrinks to fit
Undocked, the input area can shrink to its buttons; no 30% floor.

Dock again
Docking restores room for her portrait above the buttons.

Keyboard
Arrows, Shift+arrows, Home, End and double-click on the divider.

Short windows
The composer scrolls inside itself; chat history stays visible.

Narrow screens
A 640-pixel workspace pans; the page and dialogs stay anchored.

Evidence
18 composer and 38 viewport checks passed, visibly, on October 10.

◆ CARRIED FROM v1.78.0

She repaired one of her own agents

Image-Interpreter now reads a comma-separated list of images, a change credited to Tlamatini herself.

Commit 0e6fc72d is titled “Tlamatini's self-repair Image-Interpreter, by Tlamatini's-AutoBot” and is the commit the v1.78.0 tag points to. When the images field holds several paths separated by commas, each entry is resolved on its own, the results are merged, and repeated paths are removed while the original order is kept.

The list is split only when the whole value is not itself an existing file, so a single image whose name contains a comma still resolves. An entry that matches nothing is logged as a warning and the other images are still analyzed. The 22-line change arrived without a new automated test; it ships in the published v1.78.0 release.

Several images
Comma-separated paths are resolved one by one and merged.

No duplicates
Repeated paths are removed; the original order is kept.

Commas in names
A single existing file is never split at its comma.

Honest warning
An entry that resolves to nothing is logged, not hidden.

Self-repair
The commit credits Tlamatini's-AutoBot, her own self-modify work.

In v1.78.0
The tag points to this commit; no dedicated test yet.

◆ CARRIED FROM v1.77.0

She speaks to you, by name

Tagged and published on October 9; v1.78.0 replaced it as the Latest release on October 10.

Every fixed line Tlamatini sends to the chat, and that her avatar reads aloud, is now written in the first person and addressed to the user by name: “I'm ready, Angela! You can start chatting with me now.” replaces “Your agent is ready.” The lines live in one place, `agent/constants.py`; a helper fills in the first name, or the user name when there is no first name, and leaves the name out cleanly when there is neither.

Some of these lines are recognised by their words: the chat page uses them to know it is busy or ready, the avatar to decide what to say, and the Telegram bridge to filter its replies. Every one of those matchers changed in the same commit and still accepts the old wording, so a saved chat history and an older server still read correctly.

The uninstaller no longer erases the user's own work. Uninstalling an installation had deleted the projects kept in `Templates/`, where STM32er, ESP32er, Arduiner, ESPHomer, Unrealer and LaTeXer create their projects by default. It now keeps `application`, `applications`, `content_generated`, `context_files`, `doc_generated` and `Templates` whenever they hold a file, always erases `Temp/`, and still keeps `agents/`. Only a rebuilt Uninstaller.exe carries this: a fresh install or a reinstall copies it, while an in-app update keeps the old one.

Both changes are covered by unit tests (the message tests in `test_compact_mode_switch.py` and `test_uninstaller_mechanics.py`), recorded as passing in the October 9 documentation refresh.

First person
Fixed chat lines say “I” and use your name.

One source
Every fixed line lives in agent/constants.py.

Old wording kept
Matchers accept both, so old histories still read.

Templates kept
Uninstalling keeps the projects in Templates/.

Temp erased
Scratch goes; folders holding your files stay.

Rebuilt uninstaller
A reinstall carries the fix; an in-app update does not.

◆ CARRIED FROM v1.76.0

The Model Brain tunes every model

Tagged on October 8; the Model Brain commit lands on main one commit after the tag. The published v1.76.0 release was built from main and carries it.

Every model Tlamatini calls now receives its own sampling, thinking level and context size from formal sources, as the How She Works chapter describes. Config ▸ Models ▸ Save opens an Auto-tuning dialog that tunes each configured model in front of you, thinking models keep their own reasoning between tool calls inside one answer, and the stall clocks that threw finished work away are gone.

The console window paints each line by level: errors red, fatal errors white on red, warnings yellow, debug grey and successes green, while `tlamatini.log` stays plain text. At startup she logs which program draws the window, the classic console or Windows Terminal, the default on Windows 11. `console_colors` and the standard `NO_COLOR` variable turn the colours off.

Two smaller repairs ship beside it. A PDF refused because a second account signed in from the same browser now names that cause and the fix, reloading with F5, and the progress dialog marks the failed step. Two chat pages on two different models no longer make each other measure again forever: a model's verdict updates the boxes and stops. On two idle test pages the re-sends fell from 1,081 in one minute to none.

On October 8 the visible Model Brain test passed 25 of 25 checks in a source run and the visible PDF-account test passed 20 of 20. These are dated results, not a claim that this documentation refresh reran them. The Release Identity section uses the same release version as the entire dossier.

Model Brain
Per-model settings from cited sources; Auto-tuning shows each one.

Reasoning kept
Thinking models keep their train of thought between tool calls.

Coloured console
Lines coloured by level; the log file stays plain text.

Truthful refusal
A PDF refused for a second account says why and how to fix it.

Quiet tabs
Two pages on two models no longer re-measure each other.

Evidence
25 of 25 and 20 of 20 visible checks on October 8.

◆ CARRIED FROM v1.76.0

Documents and files that tell the truth

October 7 and 8: LaTeXer, Mover and the document agents report what really happened.

LaTeXer reports a PDF only when the delivered file really exists and is not empty, and a PDF whose text sticks out more than five points past the margin is reported as created with findings, not as clean. When it had to cut a quarter or more of a broken document's body to produce any PDF at all, it saves the result as `<name>.DEGRADED.pdf` and leaves the requested name untouched. In a document longer than 8,000 characters the model rung first repairs only the lines around the compiler's error with a capped request; an unusable reply falls back to the whole-document request, and a reply that never came forbids the bisect rung.

Mover no longer erases a destination folder to copy into it: folders are merged, each file is staged and published whole, a move removes its source only afterwards, and a failure ends with a failure receipt and a non-zero exit instead of a success. Document and capture agents keep a finished file when Windows Controlled Folder Access blocks the folder, and the security whitelist script now finds every Tlamatini installation and lets its Python save files.

The October 8 LaTeXer change added 37 tests, and all 549 LaTeXer tests passed with it. The October 7 Mover and LaTeXer delivery work passed 126 focused tests.

Real PDFs
Success only when the delivered file exists and is not empty.

Layout findings
Text past the margin is reported, never called clean.

DEGRADED name
A mostly-cut build never wears the name you asked for.

Region repair
Large documents: the lines around the error are repaired first.

Mover
Destinations are merged, never erased; failures are reported.

Protected folders
Finished files survive Controlled Folder Access.

◆ IN v1.76.0 · OCTOBER 5 AND 8

Smaller fixes that keep the chat moving

Committed after the v1.75.0 tag and carried by the v1.76.0 tag.

The Files-Search and System-Metrics helpers ask Ollama their own questions before the main answer. Those calls now have a time limit, `context_sidecar_llm_timeout_seconds`, 60 seconds by default; at the limit, or when you press Cancel, a helper skips its context and the answer goes ahead, so a slow reply can no longer freeze the chat.

When Windows still holds `Uninstaller.exe` while the uninstaller finishes, the file is retried, moved aside and deleted by a hidden cleanup after exit, so the uninstall completes. File-Creator gained `append`: true adds the content to the end of a file, byte for byte, instead of overwriting it.

Context helpers
Their own model calls stop at a time limit or on Cancel.

Uninstaller
A held Uninstaller.exe is moved aside and removed after exit.

File-Creator
`append: true` adds to a file instead of overwriting it.

◆ IN v1.76.0 · OCTOBER 4

Graphical notes and resizable Run output

Committed to main on October 4 and carried by the v1.76.0 tag; local executables were rebuilt to validate them.

User Commentary is edited entirely on the canvas. Double-click or press Enter to write, then format selected words with the floating mini toolbar. A single note can combine different fonts, sizes, text colors, bold, italic and underline. A caret styles the next text you type. Bubble color and alignment apply to the whole note.

Drag any border or corner, including while writing. Fit text removes spare height. Every wrapped line stays inside the bubble without an internal scrollbar. Done or Ctrl+Enter saves the edit. Cancel or Escape restores its text, styles, size and position.

The divider above Run output allocates 5% to 95% of the usable canvas/output height. Only the two viewports change. Canvas zoom, figure dimensions and text sizes stay fixed, and each pane scrolls independently. The browser remembers the ratio per signed-in user.

The October 4 commentary verification passed 29 diagram/runner tests, 3 carriage tests and 23 visible Chrome checkpoints, including three mixed-style comments through three file save/open cycles with identical data and rendered line boxes. The output-divider verification separately passed 14 visible Chrome checkpoints and 3 carriage tests. These are dated results, not a claim that this documentation refresh reran them.

◆ IN v1.76.0 · OCTOBER 3

Four stop sequences and a clearer Prompt Flow Panel

Committed on main after the v1.75.0 tag and carried by the v1.76.0 tag.

Ollama's cloud models refuse a request that carries more than four stop sequences, and the chat model used to send nine, so the chat-history summary, the first call once a conversation grows long, failed the whole request. Every stop list now passes through one fitter that keeps at most four, most important first, and a failed summary no longer fails the answer: the request continues without it, while Cancel still stops it.

The Prompt Flow Panel separates two things that shared one name. User Input is the operation that stops to ask you something; it keeps exactly its old behaviour under a new notched figure. User Commentary is now a static speech-bubble note for reviewers: written in place, coloured and styled freely, growing to hold its whole text without a scrollbar, and never run. New files are saved as version 2; version 1 files open with their old commentary steps turned into User Input, identifiers and connections kept.

The October 3 refresh reran 430 targeted unit tests, the offline Prompt Flow check, ESLint with zero errors, the skills inventory and both inclusion sweeps; all passed, and ruff now reports a clean tree. In a visible Chrome on the real desktop, the commentary test passed all nine checkpoints: long notes without scrollbars, fonts and colours, resize with Undo and Redo, save and reopen, a real User Input run, Escape stopping it, and a version 1 file migrating.

Four stop sequences
Every Ollama request; a failed summary never fails the answer.

User Input
The step that asks you: same mechanism, new figure.

User Commentary
A static review note; it never runs and never reaches the model.

Version 2 files
Version 1 flows migrate on open; identifiers and links are kept.

October 3 evidence
430 unit tests; 9 of 9 visible Chrome checkpoints.

◆ CARRIED FROM v1.75.0

Compact mode and Self-modify become switches

Tagged on October 3: Compact mode is a toolbar switch; the Self-modify switch followed and ships in v1.76.0.

Compact mode as tagged in v1.74.0 was a verdict made behind the user's back. In v1.75.0 it is a toolbar box. Ticking it unticks every row in Configure MCPs, Configure Agents and ACPX-Skills except System-Metrics, Files-Search and Current-Time, and pauses the External MCPs; unticking it ticks every row again. Each row shows the tokens it adds, a model that cannot hold everything turns the box on and locks it, and an answer built from a cut request carries a CONTEXT-WINDOW warning.

Right after the tag, commit 70aeeb87 added the Self-modify box, which ships in v1.76.0. A source run always shows it; a frozen build shows it only when built with --self-modify. On, it sends her self-knowledge, about 28,900 tokens per request; off, one line tells her not to change her own code. A model that cannot hold it shows the box locked off.

The visible Compact-switch run passed 57 of 57 checks on a local qwen2.5 and a free cloud model. The visible Self-modify run passed 28 of 28: with the same 119 tools, the request with the self-knowledge was 117,517 bytes larger than without it. 473 unit tests passed, including a render of the page as a frozen build.

Compact switch
Only the rows you tick; locked on when the model cannot hold everything.

Prices
Every Configure row shows the tokens it adds to each request.

Self-modify
Her self-knowledge on or off; locked off when it does not fit.

Proof
57/57 and 28/28 visible checks; 473 unit tests.

◆ CARRIED FROM v1.74.0

Compact mode and readable tables

Tagged on October 1: every request is fitted to the model it is sent to.

A model that can hold Tlamatini's complete request still receives it byte for byte. A model whose real window is smaller receives Compact mode, described in the How She Works chapter, with a once-per-model dialog, a Compact badge and a locked ACPX box, replaced in v1.75.0 by a switch.

The visible run on a local qwen2.5 exposed three more problems, fixed in the same release. In Compact mode the one-shot file guard now states the real reason a file cannot be opened, and a table the user asked to see is shown as a table instead of fenced code. For every model, an answer-table cell whose text would be unreadable is recoloured.

On October 1 the visible Compact run on the installed qwen2.5:latest passed 37 of 37 checks at about 8,100 tokens per request, with no cuts. On nemotron-3-ultra:cloud every request stayed complete, with Ollama reading 102,539 to 104,259 tokens. A small model can still sometimes re-answer an earlier question before the new one. The v1.74.0 dossier refresh reran the 43 Compact-mode regression tests; all passed.

Fitted
Full when the model can hold the request; Compact when it cannot.

Compact proof
37/37 visible checks on a local qwen2.5, about 8.1K tokens, no cuts.

Full proof
A large cloud model read every request complete: about 103K tokens.

Readable tables
Cells below 3:1 contrast are recoloured, for every model.

◆ CARRIED FROM v1.73.1

Model lists from the server you configured

Tagged on October 1: Config ▸ Models asks the configured Ollama servers, with their token.

Config ▸ Models used to ask the browser's own Ollama for the model list, at an address fixed when the page loaded. After Config ▸ URLs pointed Tlamatini at a remote server, a model installed only there was marked red and could not be saved.

A login-protected endpoint now reads config.json on every call, asks each distinct configured Ollama address in parallel with the same bearer token the chat sends, and merges their model names. Each server reports its own status, and the dialog names the server and the reason when one cannot answer, such as an HTTP 401 that points to the token.

The same release made the start-up GPU step send the token too, so a token-protected remote server no longer answers 401 at every start. A visible run against a simulated token-protected server passed 16 of 16 checks, with every request carrying the token. The v1.74.0 dossier refresh reran the 18 model-catalog regression tests; all passed.

Fresh address
Read from config.json on every call, never fixed at page load.

Token
Every configured server is asked with the chat's own bearer token.

Clear errors
Each server reports its status; a 401 points at the token.

Start-up
The GPU warm-up sends the token too: no 401 at every start.

◆ CARRIED FROM v1.73.0

One card removed from future context

Tagged and published on September 30: Drop throughout chat, with confirmation and history updates.

Saved greetings, user prompts and answers carry their AgentMessage id from the first page render and subsequent WebSocket frames. The drop-message request validates that id against the signed-in conversation owner before deleting rows. A busy guard checks this connection's active run; it is not a global lock across all of the user's tabs.

Successful deletion broadcasts message-dropped with all deleted ids to the user's tabs, then schedules a context-gauge refresh. Dropping a user prompt also removes up to twenty directly following Referenced Rephrase rows, stopping at the first other row. An answer or another user's message is never included in that cleanup.

All 23 focused Drop regression tests passed on September 30, covering ownership, both message roles, neighboring rows, the history window, busy refusal, rephrases and browser/server protocol contracts. The September 30 maintainer record reports 42/42 checks in the extended visible browser harness, including a new real context count after every drop that ends lower than before; this dossier refresh reruns the focused suite without writing to the real memory graph.

Owned rows
Server deletion is scoped to the signed-in user's conversation.

Tabs and gauge
Confirmed ids remove saved cards across tabs and refresh the context estimate.

Rephrases
Prompt cleanup removes following rephrase copies and keeps the answer.

Run guard
An active answer blocks Drop on that connection. Other tabs have separate runs.

◆ CARRIED FROM v1.72.4

Web pages that cannot vanish in silence

Tagged and published on September 29: no web page loses its text without saying so.

Crawler and Googler strip a page to its words by skipping scripts, styles and similar blocks. A block that never closed, such as an omitted closing head tag or a cut-off download, used to swallow the rest of the page. Both now track open blocks with a stack, and a plain-text safety net replaces any result that kept too little, logging a warning when it does.

Two tags travel with it. v1.72.2 added deadline-bound defaults to both agents, and a CAPTCHA or bot wall is reported as blocked, never read as content. v1.72.3 made the chat avatar's lips move clearly while she speaks, never wider than her own portrait.

No silent loss
A stack of open tags replaces a counter in both agents.

Safety net
A plain reading takes over when a parse keeps too little.

Bounded defaults
v1.72.2: search and crawl budgets, cancellation and explicit outcomes.

Refusals named
A CAPTCHA or bot wall comes back as blocked, never as content.

Visible lips
v1.72.3: her lips open and close clearly, without gaping.

◆ CARRIED FROM v1.72.1

A context gauge that tells the truth

Tagged and published on September 28: the ring beside the message box shows Ollama's own count.

The ring now reports the exact request Tlamatini's main chain sends. Bytes are measured; the token figure is Ollama's own prompt_eval_count for that request, and the ceiling is the context length Ollama reports. An estimate is shown only until Ollama answers, and says so.

Between questions the ring shows what the next request will cost, recalculated whenever the state changes: page open, Clear history, Drop, Clear context, a context loaded, an answer finished or a mode toggled. It is exact in one-shot and Multi-Turn alike, per connected user, and one-shot questions no longer carry an empty System-Metrics line.

Real tokens
Ollama's prompt_eval_count, paired to its own request.

Exact in both modes
Live = at-rest prediction + the question + the Multi-Turn plan.

Main chain only
Out-of-band calls never reach the ring; each user sees their own.

Less waste
The question is sent once; the one-shot placeholder line is gone.

Proof
A visible lab passed 60 of 60 checks against tlamatini.log.

◆ CARRIED FROM v1.72.0

Direct dictation and dependency coverage

These September 27 changes were tagged v1.72.0 and published on September 28.

The chat microphone starts Whisperer directly, stops on the configured silence gate, recognizes speech and sends or leaves an editable draft as selected in Config > Mic. The dialog also saves browser-local input, gain, gate and recognition preferences. The worker uses the main application logger and creates no second console. Developer tests remain visibly executed.

The root dependency manifest and separate ESPHome manifest now account for referenced libraries across agents and source. The static guard and detailed dependency audit record coverage and runtime boundaries. They shipped in the published v1.72.0 release; the shared root-asset carrier repair that followed is committed and carried by v1.72.1.

User guide

README.md and BookOfTlamatini.md explain Mic settings, recording, cancellation and both modes.

Runtime contract

docs/chat-microphone-design.md records ownership, protocol, configuration and limits.

Dependencies

docs/dependency-coverage-audit.md records declarations, isolation and verification.

Delivery

Published as v1.72.0 and carried by v1.72.1, with the root-asset carrier repair.

◆ CARRIED RELEASE

v1.70.0: the Prompt Flow Panel

Version 1.70.0 is about designing conversations and editing flows with the same ease.

Prompt Flow Panel

Seven operations, `.fpmt` files, validated playback, isolated runs.

Panels and Config

The canvases live under Panels; every Configure dialog under Config.

A real editor

Undo and Redo, Duplicate with settings, zoom, Fit, agent search, Help.

Shared mechanics

Both canvases connect, drag, select and zoom the same way.

Busy-safe menus

Panels stays usable while she answers; one path re-arms the menus.

Context line carried

The context meter, Context Governor and context gauge of v1.65.

◆ LINEAGE

How the line arrived here

Recent tags, oldest first, read from Git.

2026-10-20	**v1.88.0**	Release v1.88.0 Terminater tightly coupled to Tlamatini, redically empowering Tlammatini.
2026-10-21	**v1.89.0**	Release v1.89.0 Enabling Tlamatini's GUI the edition of files in canvas.
2026-10-22	**v1.90.0**	Release v1.90.0 Fine-tunning Tlamatini on Interactor and Terminater.
2026-10-23	**v1.91.0**	Release v1.91.0 Implemented the configuration menu entry and all the mechnism to configure the Live Editor & Terminater parameters for Tlamatini's GUI.
2026-10-27	**v1.8.2**	Release v1.8.2 Im.
2026-10-27	**v1.91.1**	Release v1.91.1 Fix on prompting flow correction with conversation_user_id, and answering handling, and some additional GUI fixes.
2026-10-30	**v1.91.2**	Release v1.91.2 Widening max speed to typing on Live Editor and Terminater , by angelahack1.
2026-11-01	**v1.92.0**	Release v1.92.0 Integrate per-user Torrenter downloads into English and Spanish chat and flows, add XT tray controls and live tests, and fix Windows packaging and Playwright uninstall cleanup.
2026-11-01	**v1.92.1**	Release v1.92.1 Fix updates, uninstaller delivery, and self-modify packaging in both editions.
2026-11-01	**v1.92.2**	Release v1.92.2 Fix(build): protect shared Python environments in English and Spanish.
2026-11-01	**v1.92.3**	Release v1.92.3 Queue Torrenter magnets when all four slots are busy and fix Executer's Playwright path handling.
2026-11-08	**v1.92.4**	Release v1.92.4 Fix matching build-version checks and isolated release builds in both Tlamatini editions.

◆ RELEASE IDENTITY

One released version: v1.92.4

The application, user guides, PDF and PowerPoint all identify the same current release. The repository metadata records the implementation inspected for this dossier.

The latest released version is 1.92.4. The application, user guides, PDF dossier and PowerPoint deck all use this one release version.

This edition documents commit `52e97879` and the complete release, including matching release builds, dependency isolation and lossless Markdown loading.

Fact	Value
Latest released version	1.92.4
Documented commit	52e97879
Coverage	Features, architecture, usage and inventory

CHAPTER

IX

Repository Facts

The source tree measured: files, languages, effective lines, the largest files and recent history.

IN THIS CHAPTER

- The repository in numbers
- Effective lines by language
- Every language, counted
- Authored and vendored
- The largest source files
- Recent commits
- How these numbers were produced

◆ INVENTORY

The repository in numbers

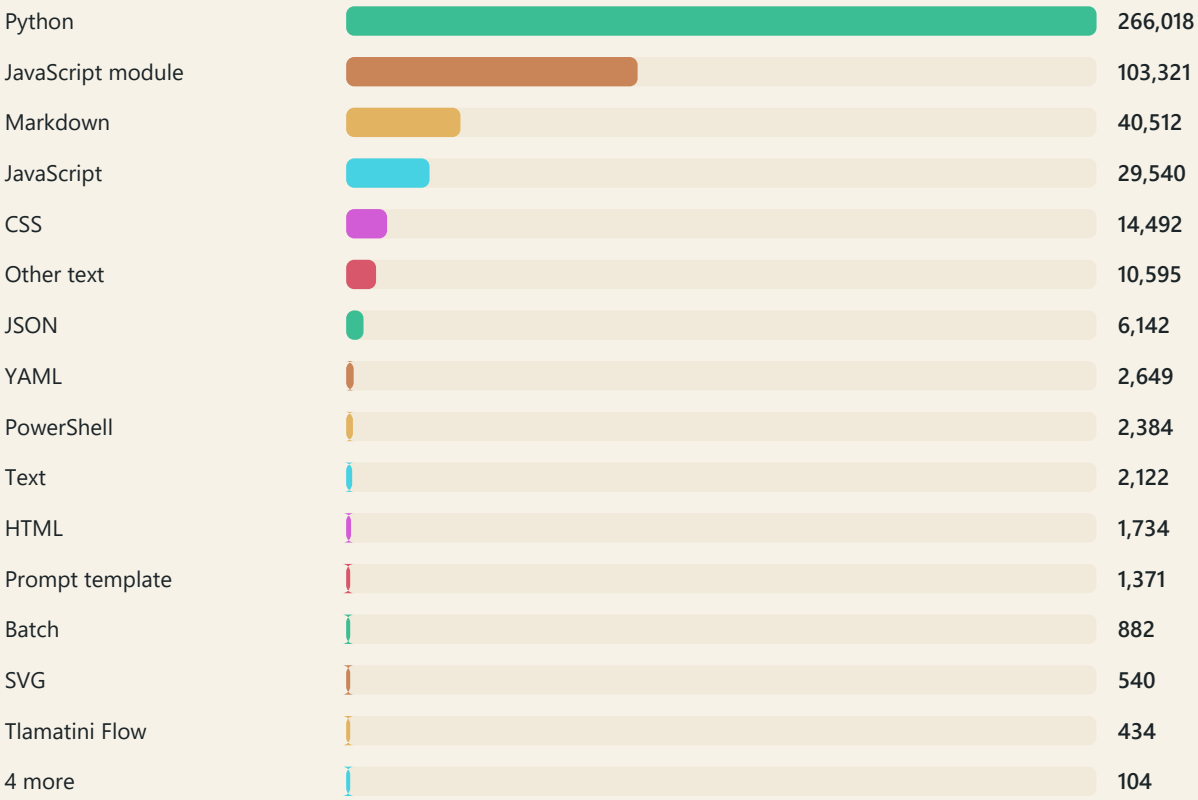
Counts come from Git and from the source files at the moment of generation.

Measure	Value
Inspected commit	52e97879 · 2026-11-08 · branch main
Version	1.92.4
Tracked files	1,860
Unignored working-tree additions	1
Text files counted	1,619
Binary assets (counted, not measured)	242
Physical text lines	617,487
Effective lines	482,840
Workflow agent types	87
Wrapped chat-agent tools	68
Built-in Multi-Turn tools	111 = 21 core + 68 wrapped + 12 ACPX/Skill + 10 supervisors
Skills	29
ACPX peers	14
Database migrations	221
JavaScript modules · stylesheets · templates	64 · 29 · 9
Pinned Python requirements	264
Catalog sections	16
Central model settings	38 for 21 model-backed agents

◆ LINE INVENTORY

Effective lines by language

482,840 effective lines across 1,619 text files; 617,487 physical lines.



◆ LINE INVENTORY

Every language, counted

Effective lines exclude blank lines, comment-only lines and Python docstrings.

Language	Files	Physical	Effective	Share
Python	938	368,594	266,018	55.1%
JavaScript module	10	104,804	103,321	21.4%
Markdown	245	53,253	40,512	8.4%
JavaScript	94	36,837	29,540	6.1%
CSS	37	17,974	14,492	3.0%
Other text	48	11,994	10,595	2.2%
JSON	14	6,142	6,142	1.3%
YAML	94	6,646	2,649	0.5%
PowerShell	20	3,057	2,384	0.5%
Text	12	2,657	2,122	0.4%
HTML	12	1,792	1,734	0.4%
Prompt template	2	1,517	1,371	0.3%
Batch	4	1,026	882	0.2%
SVG	83	574	540	0.1%
Tlamatini Flow	1	434	434	0.1%
C++	2	87	50	0.0%
Prompt flow	1	26	26	0.0%
Protocol Buffers	1	33	20	0.0%
INI	1	40	8	0.0%
Total	**1,619**	**617,487**	**482,840**	**100.0%**

◆ PROVENANCE

Authored and vendored

Third-party frontend libraries are vendored so the application needs no CDN at runtime; their lines are counted separately here so they are never mistaken for Tlamatini's own logic.

Portion	Files	Effective lines
Tlamatini-authored text (application, agents, docs, tests)	1,477	370,476
Vendored frontend libraries (PDF.js, Bootstrap, jQuery …)	142	112,364
All text files	1,619	482,840

◆ LARGEST FILES

The largest source files

Ranked by effective lines.

File	Language	Effective
Tlamatini/agent/views.py	Python	8,695
Tlamatini/agent/tests.py	Python	5,964
Tlamatini/agent/agents/flowcreator/flow_catalog.json	JSON	5,027
docs/claude/recent-fixes.md	Markdown	4,462
Tlamatini/agent/agents/latexer/latexer.py	Python	3,763
Tlamatini/agent/tools.py	Python	3,525
BookOfTlamatini.md	Markdown	3,263
Tlamatini/agent/mcp_agent.py	Python	2,683
Tlamatini/agent/doc_generation/dossier_content.py	Python	2,560
scripts/architecture_atlas/gen_architecture.py	Python	2,482
Tlamatini/agent/agents/flowcreator/agentic_skill.md	Markdown	2,371
Tlamatini/agent/chat_agent_registry.py	Python	2,292
Tlamatini/agent/external_mcp_manager.py	Python	2,144
Tlamatini/agent/static/agent/css/agentic_control_panel.css	CSS	2,067
Tlamatini/agent/static/agent/js/agent_page_chat.js	JavaScript	2,030
Tlamatini/agent/agents/stm32er/stm32er.py	Python	1,983
Tlamatini/agent/static/agent/css/agent_page.css	CSS	1,943
docs/changes/2026-10-04-commentary-graphical-redesign.patch	Other text	1,908

◆ HISTORY

Recent commits

The most recent commits on the inspected branch, newest first.

Commit	Date	Subject
52e97879	2026-11-08	Fix matching build-version checks and isolated release builds in both Tlamatini editions, by angelahck1
ca503b26	2026-11-04	Bumpuing everything as 1.92.3 latest version, by Tlamatini's-AutoBot.
13ed2f62	2026-11-02	Adding video link to the README, by angelahack1
acbc7be9	2026-11-02	Bumping to version 1.92.3, by angelahack1
74e06a6c	2026-11-01	Queue Torrenter magnets when all four slots are busy and fix Executer's Playwright path handling, by angelahack1
b99ea506	2026-11-01	Fix(build): protect shared Python environments in English and Spanish, by angelahack1
ba321572	2026-11-01	Fix updates, uninstaller delivery, and self-modify packaging in both editions, by angelahack1
21de3091	2026-11-01	Integrate per-user Torrenter downloads into English and Spanish chat and flows, add XT tray controls and live tests, and fix Windows packaging and Playwright uninstall cleanup, by angelahack1
17844abc	2026-10-30	Widening max speed to typing on Live Editor and Terminater, by angelahack1
db67eeb9	2026-10-27	Skip flow_catalog.json and 3 Terminater/canvas tests in the public scrubber
1d13922d	2026-10-27	Documentation sweep for v1.91.1 (release record, docs, skills, memory, dossiers), with keys regenerated as push-able, by Tlamatini's-AutoBot.
9915938a	2026-10-27	Fix on prompting flow correction with conversation_user_id, and answering handling, ans some additional GUI fixes, by Tñamaitni's-AutoBot.

◆ METHOD

How these numbers were produced

The inventory is reproducible: run the generator again and the same tree yields the same counts.

Scope

Every Git-tracked file plus unignored working-tree additions; ignored build output, caches and runtime state are excluded.

Text only

Binary assets such as images, fonts and archives are counted as files, never as lines.

Python

Tokenized: comments, blank lines and module, class and function docstrings are excluded.

Other languages

Block and line comments are stripped, then non-blank lines are counted.

No private values

Configuration values, credentials and addresses are never reproduced.

Regenerate

```
python Tlamatini\agent\doc_generation\refresh_project_docs.py
```

APPENDIX

A

The Application File Tree

1,638 application files, with 222 database-history files summarized. Directories are shown in jade.

◆ APPENDIX A

The application file tree

1,638 listed application files at commit `52e97879`. Within each directory, files come first and sub-directories follow; a line too long for its column continues on the next line after a » mark.

```
Tlamatini/
├── .gitattributes
├── .gitignore
├── .mcp.json
├── _acpx_pipeline_demo_report.md
├── ACPX.md
├── AGENTS.md
├── agents_descriptions.md
├── apply_update.ps1
├── audit_dependencies.py
├── AvatarImprovementByClaude.md
├── backup_db.py
├── BookOfTlamatini.md
├── build.py
├── build_all.cmd
├── build_complete_private_release.py
├── build_complete_public_release.py
├── build_env.py
├── build_installer.py
├── build_runtime_assets.py
├── build_uninstaller.py
├── check_private_data.py
├── Claude-Desktop-KALI-MCP-Session.md
├── CLAUDE.md
├── Codex2ClaudeRecommendations.md
├── ConfigureModels.png
├── copy_source_assets.py
├── CreateShortcut.json
├── CreateShortcut.ps1
├── DesignOfIncludingMemoryMCPS.txt
├── Discoverier-new-agent.md
├── eslint.config.mjs
├── example_of_prompt_to_make_de-compresser_agent.txt
├── FirstFinalPlanToSpeedUp.md
├── flow_file_associations.ps1
├── freeingport8000.ps1
├── GEMINI.md
├── git_deny_go.py
├── install.py
├── KIMI.md
├── latest-folding-context-session.md
├── LICENSE
├── MCPMemoriesFlowCreation.flw
├── MessagingConnectorAssistant_Design.md
├── ollama_credits.py
├── OllamaPricing.png
├── OOB_shift_reaper-and-NAMU-Design.pdf
├── package.json
├── PIVOT_CHANGES.md
├── pnpm-lock.yaml
├── preserved_user_state.json
├── private_targets.example.json
├── PromptDesigner.jpg
├── pyinstaller_guard.py
├── README.md
├── regen_secrets.py
├── register_flw.ps1
├── register_fpmt.ps1
├── RemoveShortcut.ps1
├── requirements-esphome.txt
├── requirements.txt
├── setup_logging.py
├── setup_recovery.py
├── setup_windows.py
├── setup_wizard.py
├── test-reloaded-session.md
├── test_author_banner.py
├── test_check_private_data.py
├── test_perf_3x_visual.py
├── test_private_data_guard.py
├── TestsVisiblesAndVisibleExecutionFromClaude2Codex.md
├── Tlamatini-Architecture.pdf
├── Tlamatini-Kali-Setup.md
├── Tlamatini.ico
├── Tlamatini.jpg
├── Tlamatini.ps1
├── tlamatini_acpx.py
├── tlamatini_app_summary.pdf
├── Tlamatini_eXtended_Artificial_Intelligence_Humanly_Tempered.pp
»   » tx
├── TLAMATINI_MCP.md
├── tlamatini_mcp_server.py
├── uncommit-keys.bat
├── uninstall.py
├── uninstall_processes.py
├── unregister_flw.ps1
├── unregister_fpmt.ps1
├── VERSIONING.md
├── versioning.py
├── .claude/
│   ├── settings.json
│   ├── hooks/
│   │   ├── announce_skills.py
│   │   └── test_announce_skills.py
│   ├── memory/
│   │   ├── feedback_agent_naming_conventions.md
│   │   ├── feedback_developer_product_boundary.md
│   │   ├── feedback_dont_overbuild_exec_safety.md
│   │   ├── feedback_forbidden_headless_visible_tests.md
│   │   ├── feedback_foreground_visible_windows.md
│   │   ├── feedback_hard_real_scenario_tests.md
│   │   ├── feedback_main_branch_only.md
│   │   ├── feedback_package_json_version_bump.md
│   │   ├── feedback_run_tlamatini_agents_visible.md
│   │   ├── feedback_state_constraints_upfront.md
│   │   ├── feedback_track_changes_pivot_file.md
│   │   ├── feedback_update_agent_docs.md
│   │   ├── feedback_user_owns_git.md
│   │   ├── MEMORY.md
│   │   ├── project_acpx_oneshot_prompt.md
│   │   ├── project_acpx_skills_menu.md
│   │   ├── project_acpx_toggle_skills.md
│   │   ├── project_acpxer_added.md
│   │   ├── project_agent_table_wiped_on_boot.md
│   │   ├── project_arduiner_agent.md
│   │   ├── project_ask_execs_feature.md
│   │   ├── project_audioplayer_agent.md
│   │   ├── project_build_concurrency_guard.md
│   │   ├── project_build_tests_and_installer_enterkey.md
│   │   ├── project_camcorder_agent.md
│   │   ├── project_carried_python_for_agents.md
│   │   ├── project_conjunction_parser_fix.md
│   │   ├── project_daily_chat_test_skill.md
│   │   ├── project_desktop_ui_lifecycle.md
│   │   ├── project_director_virtuoso_demo_prompts.md
│   │   ├── project_doc_refresh_2026_05_06.md
│   │   ├── project_doc_refresh_2026_05_08.md
│   │   ├── project_doc_refresh_2026_05_09.md
│   │   ├── project_eight_fixes_2026_05_07.md
│   │   ├── project_emailer_recmailer_oneshot.md
│   │   ├── project_embedding_memory_guard.md
│   │   ├── project_esp32er_agent.md
│   │   ├── project_execute_command_bounded_fix.md
│   │   ├── project_execute_file_foreground_fix.md
│   │   ├── project_external_exec_safety_layer.md
│   │   ├── project_flow_compiler_dialog_wins.md
│   │   ├── project_ghost_distinfo_dependency_audit.md
│   │   ├── project_kalier_agent.md
│   │   ├── project_keyboarder_wrapped_2026_05_07.md
│   │   ├── project_latexer_agent.md
│   │   ├── project_living_canvas_dropped.md
│   │   ├── project_mouser_wrapped_2026_05_07.md
│   │   ├── project_native_picker_tcltk_fix.md
│   │   ├── project_native_toast.md
│   │   ├── project_netspeed_calculator_agent.md
│   │   ├── project_numpy_pyinstaller.md
│   │   ├── project_ollama_source_build_breaks_embeddings.md
│   │   ├── project_pdfer_agent.md
```

```
│   │   ├── project_planner_followup.md
│   │   ├── project_playwrighter_agent.md
│   │   ├── project_playwrighter_hold_open.md
│   │   ├── project_prompt_catalog_mode_badges.md
│   │   ├── project_prompt_flow_commentary_input.md
│   │   ├── project_pythonxer_forkbomb_fix.md
│   │   ├── project_pythonxer_strict_ruff_gate.md
│   │   ├── project_quote_apostrophe_fix_2026_05_07.md
│   │   ├── project_reaper_console_window_fix.md
│   │   ├── project_recorder_agent.md
│   │   ├── project_reviewer_analyzer_agents.md
│   │   ├── project_reviewer_analyzer_demo_prompts.md
│   │   ├── project_reviewer_committed_secrets_falsepos.md
│   │   ├── project_secret_leak_recovery.md
│   │   ├── project_stm32er_agent.md
│   │   ├── project_stm32er_bootstrap_preflight.md
│   │   ├── project_stm32er_demo_prompts.md
│   │   ├── project_stm32er_docs_libs_parametrizer.md
│   │   ├── project_stm32er_hil_serial_proof_fix.md
│   │   ├── project_stm32er_tests.md
│   │   ├── project_summarizer_oneshot_mode.md
│   │   ├── project_taskbar_flash_attention.md
│   │   ├── project_teletlamatini_acpx.md
│   │   ├── project_temp_templates_policy.md
│   │   ├── project_unity_mcp_declined.md
│   │   ├── project_unrealer_expanded_surface.md
│   │   ├── project_versioning_2026_05_15.md
│   │   ├── project_videoplayer_agent.md
│   │   ├── project_windower_agent.md
│   │   └── user_profile.md
│   └── skills/
│       ├── tlamatini-agent-creation/
│       │   └── SKILL.md
│       ├── tlamatini-agent-naming/
│       │   └── SKILL.md
│       ├── tlamatini-daily-chat-test/
│       │   ├── .gitignore
│       │   ├── SKILL.md
│       │   └── harness/
│       │       ├── _peek.py
│       │       ├── _stage.ps1
│       │       ├── acp_backend_banner_test.py
│       │       ├── acp_backend_banner_test.py.antes_shoter
│       │       ├── acp_dialog_full_sweep.py
│       │       ├── acp_dialog_theme_visible.py
│       │       ├── ask_execs_regression.py
│       │       ├── ask_execs_regression.py.antes_shoter
│       │       ├── bulk_toggle_behaviour.py
│       │       ├── bulk_toggle_visible.py
│       │       ├── cancel_regression.py
│       │       ├── cancel_regression.py.antes_shoter
│       │       ├── compact_mode_visible.py
│       │       ├── compact_pingpong_visible.py
│       │       ├── compact_switch_visible.py
│       │       ├── config.py
│       │       ├── context_gauge_100.py
│       │       ├── context_gauge_real_lab.py
│       │       ├── context_gauge_visible.py
│       │       ├── context_restore_spinner_visible.py
│       │       ├── cyber_watchdog_test.py
│       │       ├── deepseek_remote_chat_visible.py
│       │       ├── dialog_policy_visible.py
│       │       ├── dialog_theme_visible.py
│       │       ├── discoverer_1000.py
│       │       ├── discoverer_1000.py.antes_shoter
│       │       ├── drop_message_visible.py
│       │       ├── file_drop_visible.py
│       │       ├── flowcreator_wizard_e2e.py
│       │       ├── full_mode_visible.py
│       │       ├── googler_dork_hunt.py
│       │       ├── grepper_lines_visible.py
│       │       ├── grepper_login_probe.py
│       │       ├── latexer_openmp_e2e.py
│       │       ├── latexer_stepbystep_5.py
│       │       ├── mcp_playwright_suite.py
│       │       ├── model_brain_visible.py
│       │       ├── monitor.py
│       │       ├── ollama_remote_catalog_visible.py
│       │       ├── pdf_context_visible.py
│       │       ├── pdfer_nuance_visible.py
│       │       ├── playwrighter_run.py
│       │       ├── pptxer_visible.py
│       │       ├── preflight.py
│       │       ├── qualify.py
│       │       ├── questions.py
│       │       ├── README.md
│       │       ├── requirements.txt
│       │       ├── run_test.py
│       │       ├── self_modify_visible.py
│       │       ├── shoter_shot.py
│       │       ├── step_by_step_catalog.py
│       │       ├── step_by_step_catalog.py.antes_shoter
│       │       ├── stop_sequence_visible.py
│       │       ├── talk_test.py
│       │       ├── uninstaller_visible.py
│       │       ├── video_link_visible.py
│       │       ├── voice_commands_visible.py
│       │       ├── window-test-screen.py
│       │       ├── windower_focus.py
│       │       └── wrapped_questions.py
│       ├── tlamatini-self-modify-inclusion/
│       │   ├── SKILL.md
│       │   └── scripts/
│       │       └── sweep_self_modify.py
│       └── tlamatini-self-update-inclusion/
│           ├── SKILL.md
│           └── scripts/
│               └── sweep_self_update.py
├── .codex/
│   └── skills/
│       ├── full-project-pdf-dossier/
│       │   └── SKILL.md
│       ├── overlap-safe-pptx-dossier/
│       │   └── SKILL.md
│       └── tlamatini-release-validation/
│           ├── SKILL.md
│           ├── agents/
│           │   └── openai.yaml
│           └── references/
│               └── acceptance-matrix.md
├── .gemini/
│   └── skills/
│       ├── tlamatini-agent-creation/
│       │   └── SKILL.md
│       ├── tlamatini-agent-naming/
│       │   └── SKILL.md
│       ├── tlamatini-daily-chat-test/
│       │   ├── .gitignore
│       │   ├── SKILL.md
│       │   └── harness/
│       │       ├── acp_backend_banner_test.py
│       │       ├── acp_backend_banner_test.py.antes_shoter
│       │       ├── acp_dialog_full_sweep.py
│       │       ├── acp_dialog_theme_visible.py
│       │       ├── ask_execs_regression.py
│       │       ├── ask_execs_regression.py.antes_shoter
│       │       ├── bulk_toggle_behaviour.py
│       │       ├── bulk_toggle_visible.py
│       │       ├── cancel_regression.py
│       │       ├── cancel_regression.py.antes_shoter
│       │       ├── config.py
│       │       ├── cyber_watchdog_test.py
│       │       ├── dialog_policy_visible.py
│       │       ├── dialog_theme_visible.py
│       │       ├── discoverer_1000.py
│       │       ├── discoverer_1000.py.antes_shoter
│       │       ├── flowcreator_wizard_e2e.py
│       │       ├── googler_dork_hunt.py
│       │       ├── latexer_openmp_e2e.py
│       │       ├── latexer_stepbystep_5.py
│       │       ├── mcp_playwright_suite.py
│       │       ├── playwrighter_run.py
│       │       ├── qualify.py
│       │       ├── questions.py
│       │       ├── README.md
│       │       ├── requirements.txt
│       │       ├── run_test.py
│       │       ├── shoter_shot.py
│       │       ├── step_by_step_catalog.py
│       │       ├── step_by_step_catalog.py.antes_shoter
│       │       ├── talk_test.py
│       │       └── wrapped_questions.py
│       ├── tlamatini-self-modify-inclusion/
│       │   ├── SKILL.md
│       │   └── scripts/
│       │       └── sweep_self_modify.py
│       └── tlamatini-self-update-inclusion/
│           ├── SKILL.md
│           └── scripts/
│               └── sweep_self_update.py
├── .github/
│   └── workflows/
│       └── name-guard.yml
├── .scanning/
│   └── finding-policy.json
├── AuxTests/
│   └── mypdf2.py
└── docs/
    ├── agent-coverage.md
    ├── build-root-assets-verification.md
    ├── canvas-and-interactor-quick-reference.txt
    ├── canvas-editing.md
    ├── canvas-refresh.md
    ├── chat-microphone-design.md
    ├── combined-release-builder.md
    ├── companion-app-discovery.md
    ├── dependency-coverage-audit.md
    ├── desktop-input-and-flow-contracts.md
```

```
    ├── external_mcp_bulletproof_architecture.md
    ├── GUI-Manager-design.md
    ├── interaction-settings.md
    ├── interactor.md
    ├── markdown-canvas.md
    ├── model-configuration-verification.md
    ├── model_configuration.md
    ├── oob-namu-lifecycle.md
    ├── pdf-canvas.md
    ├── prompting-flow-designer.md
    ├── self-management-carriage.md
    ├── setup-lifecycle-research.md
    ├── setup-wizards.md
    ├── stm32er_all_families_proposal.md
    ├── terminater.md
    ├── torrenter.md
    ├── usage-dashboard.md
    ├── user-creation-wizard.md
    ├── visual-analysis-errors.md
    ├── windows-flow-files.md
    ├── assets/
    │   └── readme-hero.svg
    ├── changes/
    │   ├── 2026-10-03-prompt-commentary-input.md
    │   ├── 2026-10-03-prompt-commentary-input.patch
    │   ├── 2026-10-03-prompt-flow-documentation.md
    │   ├── 2026-10-03-prompt-flow-documentation.patch
    │   ├── 2026-10-04-commentary-graphical-redesign.md
    │   ├── 2026-10-04-commentary-graphical-redesign.patch
    │   ├── 2026-10-04-prompt-flow-documentation.md
    │   ├── 2026-10-04-release-validation.md
    │   ├── 2026-10-04-run-output-resize.md
    │   ├── 2026-10-04-run-output-resize.patch
    │   ├── 2026-10-04-windows-flow-files.md
    │   ├── 2026-10-07-latexer-verification.md
    │   ├── 2026-10-07-pdf-mover-incident.md
    │   ├── 2026-10-07-windows-access-verification.md
    │   ├── 2026-10-10-composer-resize.md
    │   ├── 2026-10-11-ui-sweep.md
    │   ├── 2026-10-13-docked-avatar-divider.md
    │   ├── 2026-10-16-v1.85.0.md
    │   ├── 2026-10-18-v1.87.0.md
    │   ├── 2026-10-19-private-release-data.md
    │   ├── 2026-10-19-terminater.md
    │   ├── 2026-10-20-terminater-audit.md
    │   ├── 2026-10-20-v1.88.0.md
    │   ├── 2026-10-22-documentation-refresh.md
    │   ├── 2026-10-22-release-version-pinning.md
    │   ├── 2026-10-22-v1.90.0.md
    │   ├── 2026-10-23-v1.91.0.md
    │   ├── 2026-10-27-v1.91.1.md
    │   └── 2026-11-01-v1.92.3.md
    ├── claude/
    │   ├── acpx.md
    │   ├── agents.md
    │   ├── architecture.md
    │   ├── exec-report.md
    │   ├── frontend.md
    │   ├── gotchas.md
    │   ├── INDEX.md
    │   ├── mcp-tools.md
    │   ├── multi-turn.md
    │   └── recent-fixes.md
    └── examples/
        └── prompting-kickoff.fpmt
├── pyinstaller_hooks/
    ├── hook-numpy.py
    ├── hook-pymupdf.py
    └── hook-torch.py
├── scripts/
    ├── acp_editor_visible.py
    ├── agent_catalog_visible.py
    ├── build_canvas_editor.mjs
    ├── build_setup_full_payload.py
    ├── build_setup_validation.py
    ├── canvas_editing_visible.py
    ├── canvas_editor_entry.mjs
    ├── canvas_refresh_visible.py
    ├── canvas_text_decoding_visible.py
    ├── chat_composer_resize_checks.py
    ├── chat_composer_resize_visible.py
    ├── chat_composer_visible_window.py
    ├── chat_microphone_visible.py
    ├── chat_viewport_visible_checks.py
    ├── chat_views_frozen_visible.py
    ├── chat_views_visible.py
    ├── check_js_parse.mjs
    ├── check_python_quality.py
    ├── check_requirements_coverage.py
    ├── docked_avatar_resize_checks.py
    ├── dossier_visual_walkthrough_visible.py
    ├── drag_divider_look_visible.py
    ├── file_drop_router_cases.mjs
    ├── flow_canvas_mechanics_visible.py
    ├── flow_file_associations_test.ps1
    ├── flow_files_visible.py
    ├── installed_panels_visible.py
    ├── interaction_settings_visible.py
    ├── interactor_runtime_server.py
    ├── interactor_runtime_visible.py
    ├── interactor_visible.py
    ├── markdown_canvas_visible.py
    ├── menu_browser_checks.py
    ├── menu_live_checks.py
    ├── menu_reference_graph.mjs
    ├── panel_search_title_visible.py
    ├── playwright_real_visibility.cjs
    ├── pptxer_visibility_gallery.py
    ├── prompt_flow_commentary_visible.py
    ├── prompt_flow_connections_visible.py
    ├── prompt_flow_extension_visible.py
    ├── prompt_flow_output_resize_visible.py
    ├── prompt_flow_selection_visible.py
    ├── refresh_supporting_pdfs.py
    ├── release_acp_editor_checks.py
    ├── release_acp_runtime_checks.py
    ├── release_agent_catalog_checks.py
    ├── release_auth_checks.py
    ├── release_commentary_checks.py
    ├── release_extended_checks.py
    ├── release_local_agent_checks.py
    ├── release_mcp_checks.py
    ├── release_output_resize_checks.py
    ├── release_prompt_runtime_checks.py
    ├── release_shell_file_checks.py
    ├── review_documentation_visuals.py
    ├── run_menu_state_checks.py
    ├── setup_crash_agents_visible.py
    ├── setup_desktop_cycles.py
    ├── setup_lifecycle_checks.py
    ├── setup_release_cycles.py
    ├── setup_wizard_visible.py
    ├── skill_inventory.py
    ├── sync_assistant_skills.py
    ├── terminater_visible_acp.py
    ├── terminater_visible_all.py
    ├── terminater_visible_battery.py
    ├── terminater_visible_chat.py
    ├── terminater_visible_common.py
    ├── terminater_visible_pfp.py
    ├── tlamatini_clean_install.ps1
    ├── ui_sweep_resize_checks.py
    ├── ui_sweep_visible.py
    ├── update_flow_catalog.py
    ├── usage_dashboard_browser_checks.py
    ├── usage_dashboard_fixture_checks.py
    ├── usage_dashboard_visible.py
    ├── vendor_frontend.py
    ├── vendor_pdfjs.py
    ├── verify_latexer_styles.py
    ├── verify_pdfer_styles.py
    ├── verify_pptxer_layout.py
    ├── verify_pptxer_styles.py
    ├── visual_fatal_dialog_visible.py
    └── architecture_atlas/
        ├── build_architecture_pdf.py
        ├── gen_architecture.py
        └── README.md
├── security/
    ├── automated_tests_of_security_assets.py
    ├── enable_tlamatini_v2.bat
    ├── README.md
    ├── run_defender.bat
    ├── sync_enable_launcher.py
    ├── test_security_syntax.ps1
    ├── test_windows_access.ps1
    ├── tlamatini_defender.ps1
    ├── tlamatini_whitelist_v2.ps1
    ├── verify_agent_execution.py
    └── windows_access_helpers.ps1
├── Tests/
    ├── run_avatar_tests.py
    ├── run_presence_tests.ps1
    ├── test_about_window_name_visible.py
    ├── test_avatar_animation.cjs
    ├── test_avatar_collectstatic.py
    ├── test_avatar_presence_visible.py
    ├── test_avatar_visible.cjs
    ├── test_build_env_concurrency.py
    ├── test_canvas_text_decoding.cjs
    ├── test_pdf_canvas_assets.py
    ├── test_pdf_canvas_browser.py
    ├── test_perf_3x_visual.py
    ├── test_release_regeneration.py
    ├── test_release_type_banner.py
    ├── test_release_version_pinning.py
    ├── test_requirements_coverage.py
    ├── test_setup_release_paths.py
```

```
│   ├── test_update_client_carriage.py
│   └── avatar_dev_server/
│       ├── dev_settings.py
│       └── development_test.py
└── Tlamatini/
    ├── cat_art.py
    ├── manage.py
    ├── .agents/
    │   └── workflows/
    │       └── create_new_agent.md
    ├── .mcps/
    │   └── create_new_mcp.md
    ├── .skills/
    │   └── create_new_skill.md
    ├── agent/
    │   ├── __init__.py
    │   ├── access_key_wizard.py
    │   ├── admin.py
    │   ├── agent_manifest.py
    │   ├── agent_verdict.py
    │   ├── apps.py
    │   ├── cancellation.py
    │   ├── canvas_file_views.py
    │   ├── capability_registry.py
    │   ├── chain_files_search_lcel.py
    │   ├── chain_system_lcel.py
    │   ├── chat_agent_registry.py
    │   ├── chat_agent_runtime.py
    │   ├── chat_history_loader.py
    │   ├── chat_voice_consumer.py
    │   ├── chat_voice_runtime.py
    │   ├── chat_voice_settings.py
    │   ├── command_togen_pb2.txt
    │   ├── command_watchdog.py
    │   ├── compact_mode.py
    │   ├── config.json
    │   ├── config_loader.py
    │   ├── constants.py
    │   ├── consumers.py
    │   ├── contacts.py
    │   ├── context_baseline.py
    │   ├── context_fitter.py
    │   ├── context_governor.py
    │   ├── context_sidecar_timeout.py
    │   ├── embedding_memory_guard.py
    │   ├── exec_permission.py
    │   ├── external_mcp_defaults.py
    │   ├── external_mcp_manager.py
    │   ├── external_mcps.json
    │   ├── filesearch.proto
    │   ├── filesearch_pb2.py
    │   ├── filesearch_pb2_grpc.py
    │   ├── flow_file_open.py
    │   ├── flow_file_views.py
    │   ├── global_execution_planner.py
    │   ├── global_state.py
    │   ├── gpu_perf.py
    │   ├── inet_determiner.py
    │   ├── interaction_narration.py
    │   ├── interaction_settings.py
    │   ├── llm_timing.py
    │   ├── local_toolcall_parser.py
    │   ├── log_identity.py
    │   ├── mcp_agent.py
    │   ├── mcp_files_search_client.py
    │   ├── mcp_files_search_server.py
    │   ├── mcp_system_client.py
    │   ├── mcp_system_server.py
    │   ├── model_brain.py
    │   ├── model_brain_chat.py
    │   ├── model_brain_views.py
    │   ├── model_profiles.json
    │   ├── models.py
    │   ├── native_dialogs.py
    │   ├── ollama_stop.py
    │   ├── orphan_reaper.py
    │   ├── path_guard.py
    │   ├── pdf_context.py
    │   ├── pdf_context_jobs.py
    │   ├── pdf_context_views.py
    │   ├── pdf_image_analysis.py
    │   ├── prompt.pmt
    │   ├── prompt_flow_panel_consumer.py
    │   ├── prompt_flow_panel_runtime.py
    │   ├── rag_enhancements.py
    │   ├── routing.py
    │   ├── run_db_wal_tests.ps1
    │   ├── runtime_provisioner.py
    │   ├── self_healing.py
    │   ├── self_update.py
    │   ├── sqlite_copy.py
    │   ├── terminal_consumer.py
    │   ├── terminal_guard.py
    │   ├── terminal_hub.py
    │   ├── terminal_views.py
    │   ├── terminater_testkit.py
    │   ├── terminater_tool.py
    │   ├── test_access_key_wizard.py
    │   ├── test_agent_display_names.py
    │   ├── test_agent_manifest.py
    │   ├── test_agent_verdict.py
    │   ├── test_agents_see_hidden_dirs.py
    │   ├── test_arduiner_agent.py
    │   ├── test_ask_execs_allowlist.py
    │   ├── test_audioplayer_agent.py
    │   ├── test_binary_guard.py
    │   ├── test_blenderer_agent.py
    │   ├── test_build_env.py
    │   ├── test_build_pip_quiet.py
    │   ├── test_build_root_assets.py
    │   ├── test_build_scripts.py
    │   ├── test_camcorder_agent.py
    │   ├── test_cancellation.py
    │   ├── test_canvas_file_views.py
    │   ├── test_catalog_pruning.py
    │   ├── test_chain_readiness.py
    │   ├── test_chat_agent_run_wait.py
    │   ├── test_chat_bridge_bots.py
    │   ├── test_chat_history_window.py
    │   ├── test_chat_voice.py
    │   ├── test_checkbox_bulk_toggle.py
    │   ├── test_command_watchdog.py
    │   ├── test_compact_mode_pingpong.py
    │   ├── test_compact_mode_switch.py
    │   ├── test_configure_dialog_initial_ticks.py
    │   ├── test_console_colors.py
    │   ├── test_console_shield.py
    │   ├── test_contacts.py
    │   ├── test_context_governor.py
    │   ├── test_context_real_tokens.py
    │   ├── test_context_restore_spinner.py
    │   ├── test_context_sidecar_timeout.py
    │   ├── test_crawler_agent.py
    │   ├── test_crawler_resilience.py
    │   ├── test_createsuperuser_wizard.py
    │   ├── test_ctrl_c_shutdown.py
    │   ├── test_db_backup_restore_wal.py
    │   ├── test_db_backup_safety.py
    │   ├── test_de_compresser.py
    │   ├── test_deep_research_prompt.py
    │   ├── test_deleter_safety.py
    │   ├── test_desktop_input_agents.py
    │   ├── test_dialog_dismissal_policy.py
    │   ├── test_dialog_theme.py
    │   ├── test_discoverer_agent.py
    │   ├── test_discoverer_thousand.py
    │   ├── test_django_port_config.py
    │   ├── test_drag_divider_look.py
    │   ├── test_drop_message.py
    │   ├── test_editor_agent.py
    │   ├── test_embedding_memory_guard.py
    │   ├── test_esp32er_agent.py
    │   ├── test_esphomer_agent.py
    │   ├── test_exec_report_verdict.py
    │   ├── test_execute_command_bounded.py
    │   ├── test_executer_shell_request.py
    │   ├── test_executer_window_report.py
    │   ├── test_external_mcp_add_flow.py
    │   ├── test_external_mcp_e2e.py
    │   ├── test_external_mcp_transports.py
    │   ├── test_external_mcp_universal.py
    │   ├── test_file_creator_append.py
    │   ├── test_file_drop_router.py
    │   ├── test_file_extractor_reader.py
    │   ├── test_flow_contracts.py
    │   ├── test_flow_file_open.py
    │   ├── test_flow_file_views.py
    │   ├── test_flow_knowledge.py
    │   ├── test_frontend_mutable_state.py
    │   ├── test_gitter_custom_command.py
    │   ├── test_glob_patterns.py
    │   ├── test_globber_agent.py
    │   ├── test_googler_agent.py
    │   ├── test_googler_dorks.py
    │   ├── test_googler_resilience.py
    │   ├── test_grepper_agent.py
    │   ├── test_grepper_encodings.py
    │   ├── test_grepper_lines_mode.py
    │   ├── test_image_interpreter_agent.py
    │   ├── test_instant_messaging_doctor.py
    │   ├── test_interaction_settings.py
    │   ├── test_interactor.py
    │   ├── test_j_decompiler_agent.py
    │   ├── test_js_gates.py
    │   ├── test_kalier_agent.py
    │   ├── test_latexer_agent.py
    │   ├── test_latexer_delivery_truth.py
    │   ├── test_latexer_layout_truth.py
```

```
    ├── test_latexer_major_removal_and_window_repair.py
    ├── test_latexer_model_repair_truth.py
    ├── test_latexer_repair_ladder.py
    ├── test_latexer_styles.py
    ├── test_latexer_suite.py
    ├── test_latexer_verbatim_channel.py
    ├── test_log_identity.py
    ├── test_mcp_configured_endpoints.py
    ├── test_model_brain.py
    ├── test_model_capacity.py
    ├── test_model_settings.py
    ├── test_mover_file_safety.py
    ├── test_native_dialogs.py
    ├── test_netspeed_calculator_agent.py
    ├── test_nmapper_agent.py
    ├── test_no_audio_in_tests.py
    ├── test_ollama_catalog_from_server.py
    ├── test_ollama_stop_limit.py
    ├── test_orphan_reaper.py
    ├── test_parametrizer_mcp_doctor.py
    ├── test_password_quoting.py
    ├── test_pdf_context.py
    ├── test_pdfer_agent.py
    ├── test_pdfer_nuance_layout.py
    ├── test_pdfer_styles.py
    ├── test_playwrighter_agent.py
    ├── test_pptxer_agent.py
    ├── test_pptxer_styles.py
    ├── test_pptxer_visibility.py
    ├── test_preserved_user_state.py
    ├── test_program_name_collisions.py
    ├── test_prompt_catalog_contiguous.py
    ├── test_prompt_flow_panel.py
    ├── test_prompt_flow_panel_carriage.py
    ├── test_prompt_flow_panel_runtime.py
    ├── test_prompt_flow_panel_websocket.py
    ├── test_prompt_shape_sentences.py
    ├── test_public_release_targets.py
    ├── test_pyinstaller_guard.py
    ├── test_recorder_agent.py
    ├── test_release_payload_ownership.py
    ├── test_runtime_provisioner.py
    ├── test_security_assets_carriage.py
    ├── test_self_healing.py
    ├── test_self_knowledge_is_current.py
    ├── test_self_modify_gate.py
    ├── test_self_modify_switch.py
    ├── test_self_update.py
    ├── test_shoter_takes_screenshots.py
    ├── test_sqlite_wal_mode.py
    ├── test_status_vocabulary.py
    ├── test_step_by_step_mode.py
    ├── test_stm32er_agent.py
    ├── test_talker_agent.py
    ├── test_telegrammer_identity.py
    ├── test_temp_dir_policy.py
    ├── test_terminater_agent.py
    ├── test_terminater_engine.py
    ├── test_terminater_hub_api.py
    ├── test_terminater_integration.py
    ├── test_terminater_pfp.py
    ├── test_terminater_regressions.py
    ├── test_terminater_reliability.py
    ├── test_terminater_text.py
    ├── test_tool_failure_recovery.py
    ├── test_torrenter.py
    ├── test_torrenter_delivery.py
    ├── test_uninstall_processes.py
    ├── test_uninstaller_lifecycle.py
    ├── test_uninstaller_mechanics.py
    ├── test_unrealer_agent.py
    ├── test_usage.py
    ├── test_version_guard.py
    ├── test_video_analyzer_agent.py
    ├── test_video_analyzer_content.py
    ├── test_video_analyzer_fetch.py
    ├── test_videoplayer_agent.py
    ├── test_visual_fatal_errors.py
    ├── test_voice_commands_catalog.py
    ├── test_watchdog_foreground_exemption.py
    ├── test_web_process_stays_lean.py
    ├── test_whatsapper_identity.py
    ├── test_whisperer_agent.py
    ├── test_win_shim.py
    ├── test_windower_agent.py
    ├── test_windows_app_registration.py
    ├── test_wizard_intent_guards.py
    ├── tests.py
    ├── tests_perf_3x.py
    ├── Tlamatini.md
    ├── tools.py
    ├── torrenter_bridge.py
    ├── torrenter_tool.py
    ├── torrenter_views.py
    ├── urls.py
    ├── usage_provider.py
    ├── usage_tracking.py
    ├── usage_views.py
    ├── version.py
    ├── views.py
    ├── visual_error_reporting.py
    ├── web_search_llm.py
    ├── win_shim.py
    ├── window_flash.py
    ├── windows_app_registration.py
    ├── acpx/
    │   ├── __init__.py
    │   ├── agent_registry.py
    │   ├── child_health.py
    │   ├── config.py
    │   ├── permissions.py
    │   ├── runtime.py
    │   ├── service.py
    │   ├── session_store.py
    │   ├── tests.py
    │   ├── tools.py
    │   └── windows_spawn.py
    ├── agents/
    │   ├── model_settings.py
    │   ├── visual_errors.py
    │   ├── acpxer/
    │   │   ├── acpxer.py
    │   │   └── config.yaml
    │   ├── analyzer/
    │   │   ├── analyzer.py
    │   │   └── config.yaml
    │   ├── and/
    │   │   ├── and.py
    │   │   └── config.yaml
    │   ├── apirer/
    │   │   ├── apirer.py
    │   │   └── config.yaml
    │   ├── arduiner/
    │   │   ├── arduiner.py
    │   │   ├── config.yaml
    │   │   └── ArduinoTemplateProject/
    │   │       ├── ArduinoTemplateProject.ino
    │   │       ├── README.md
    │   │       ├── sketch.yaml
    │   │       └── src/
    │   │           ├── Heartbeat.cpp
    │   │           └── Heartbeat.h
    │   ├── asker/
    │   │   ├── asker.py
    │   │   └── config.yaml
    │   ├── audioplayer/
    │   │   ├── audioplayer.py
    │   │   └── config.yaml
    │   ├── barrier/
    │   │   ├── barrier.py
    │   │   ├── config.yaml
    │   │   └── test_barrier.py
    │   ├── blenderer/
    │   │   ├── blenderer.py
    │   │   └── config.yaml
    │   ├── camcorder/
    │   │   ├── camcorder.py
    │   │   └── config.yaml
    │   ├── cleaner/
    │   │   ├── __init__.py
    │   │   ├── cleaner.py
    │   │   └── config.yaml
    │   ├── counter/
    │   │   ├── config.yaml
    │   │   └── counter.py
    │   ├── crawler/
    │   │   ├── config.yaml
    │   │   └── crawler.py
    │   ├── croner/
    │   │   ├── config.yaml
    │   │   └── croner.py
    │   ├── de_compresser/
    │   │   ├── config.yaml
    │   │   └── de_compresser.py
    │   ├── deleter/
    │   │   ├── config.yaml
    │   │   └── deleter.py
    │   ├── discoverer/
    │   │   ├── config.yaml
    │   │   └── discoverer.py
    │   ├── dockerer/
    │   │   ├── config.yaml
    │   │   └── dockerer.py
    │   ├── editor/
    │   │   ├── config.yaml
    │   │   └── editor.py
    │   ├── emailer/
    │   │   ├── config.yaml
```

```
│   │   │   └── emailer.py
│   │   ├── ender/
│   │   │   ├── config.yaml
│   │   │   └── ender.py
│   │   ├── esp32er/
│   │   │   ├── config.yaml
│   │   │   ├── esp32er.py
│   │   │   └── ESP32TemplateProject/
│   │   │       ├── platformio.ini
│   │   │       ├── README.md
│   │   │       ├── include/
│   │   │       │   └── README.md
│   │   │       ├── lib/
│   │   │       │   └── README.md
│   │   │       ├── src/
│   │   │       │   └── main.cpp
│   │   │       └── test/
│   │   │           └── README.md
│   │   ├── esphomer/
│   │   │   ├── config.yaml
│   │   │   ├── esphomer.py
│   │   │   └── ESPHomeTemplateProject/
│   │   │       ├── .gitignore
│   │   │       ├── README.md
│   │   │       ├── secrets.yaml.example
│   │   │       ├── tlamatini-light.yaml
│   │   │       ├── tlamatini-sensor.yaml
│   │   │       └── common/
│   │   │           └── base.yaml
│   │   ├── executer/
│   │   │   ├── config.yaml
│   │   │   └── executer.py
│   │   ├── file_creator/
│   │   │   ├── config.yaml
│   │   │   └── file_creator.py
│   │   ├── file_extractor/
│   │   │   ├── config.yaml
│   │   │   └── file_extractor.py
│   │   ├── file_interpreter/
│   │   │   ├── config.yaml
│   │   │   └── file_interpreter.py
│   │   ├── flowbacker/
│   │   │   ├── config.yaml
│   │   │   └── flowbacker.py
│   │   ├── flowcreator/
│   │   │   ├── agentic_skill.md
│   │   │   ├── config.yaml
│   │   │   ├── flow_catalog.json
│   │   │   ├── flow_knowledge.py
│   │   │   ├── flowcreator.py
│   │   │   └── result_to_flw.py
│   │   ├── flowhypervisor/
│   │   │   ├── config.yaml
│   │   │   ├── flowhypervisor.py
│   │   │   ├── hypervisor_alert.wav
│   │   │   └── monitoring-prompt.pmt
│   │   ├── forker/
│   │   │   ├── config.yaml
│   │   │   └── forker.py
│   │   ├── gitter/
│   │   │   ├── config.yaml
│   │   │   └── gitter.py
│   │   ├── globber/
│   │   │   ├── config.yaml
│   │   │   └── globber.py
│   │   ├── googler/
│   │   │   ├── config.yaml
│   │   │   ├── googler.py
│   │   │   └── googler_engines.py
│   │   ├── grepper/
│   │   │   ├── config.yaml
│   │   │   └── grepper.py
│   │   ├── image_interpreter/
│   │   │   ├── config.yaml
│   │   │   └── image_interpreter.py
│   │   ├── instant_messaging_doctor/
│   │   │   ├── config.yaml
│   │   │   └── instant_messaging_doctor.py
│   │   ├── j_decompiler/
│   │   │   ├── config.yaml
│   │   │   └── j_decompiler.py
│   │   ├── jenkinser/
│   │   │   ├── config.yaml
│   │   │   └── jenkinser.py
│   │   ├── kalier/
│   │   │   ├── config.yaml
│   │   │   └── kalier.py
│   │   ├── keyboarder/
│   │   │   ├── config.yaml
│   │   │   ├── keyboarder.py
│   │   │   └── keyboarder_input.py
│   │   ├── kuberneter/
│   │   │   ├── config.yaml
│   │   │   └── kuberneter.py
│   │   ├── kyber_cipher/
│   │   │   ├── config.yaml
│   │   │   └── kyber_cipher.py
│   │   ├── kyber_decipher/
│   │   │   ├── config.yaml
│   │   │   └── kyber_decipher.py
│   │   ├── kyber_keygen/
│   │   │   ├── config.yaml
│   │   │   └── kyber_keygen.py
│   │   ├── latexer/
│   │   │   ├── config.yaml
│   │   │   ├── latexer.py
│   │   │   ├── latexer_artwork.py
│   │   │   ├── latexer_design.py
│   │   │   ├── latexer_styles.py
│   │   │   └── STYLES.md
│   │   ├── mcp_doctor/
│   │   │   ├── config.yaml
│   │   │   └── mcp_doctor.py
│   │   ├── mongoxer/
│   │   │   ├── config.yaml
│   │   │   └── mongoxer.py
│   │   ├── monitor_log/
│   │   │   ├── config.yaml
│   │   │   └── monitor_log.py
│   │   ├── monitor_netstat/
│   │   │   ├── config.yaml
│   │   │   └── monitor_netstat.py
│   │   ├── mouser/
│   │   │   ├── config.yaml
│   │   │   ├── mouser.py
│   │   │   └── mouser_coordinates.py
│   │   ├── mover/
│   │   │   ├── config.yaml
│   │   │   └── mover.py
│   │   ├── netspeed_calculator/
│   │   │   ├── config.yaml
│   │   │   └── netspeed_calculator.py
│   │   ├── nmapper/
│   │   │   ├── config.yaml
│   │   │   └── nmapper.py
│   │   ├── notifier/
│   │   │   ├── config.yaml
│   │   │   └── notifier.py
│   │   ├── or/
│   │   │   ├── config.yaml
│   │   │   └── or.py
│   │   ├── parametrizer/
│   │   │   ├── config.yaml
│   │   │   └── parametrizer.py
│   │   ├── pdfer/
│   │   │   ├── config.yaml
│   │   │   ├── pdfer.py
│   │   │   ├── pdfer_artwork.py
│   │   │   ├── pdfer_atelier.py
│   │   │   ├── pdfer_audit.py
│   │   │   ├── pdfer_color.py
│   │   │   ├── pdfer_consult.py
│   │   │   ├── pdfer_docmodel.py
│   │   │   ├── pdfer_nuance.py
│   │   │   ├── pdfer_ornament.py
│   │   │   ├── pdfer_styles.py
│   │   │   ├── pdfer_tables.py
│   │   │   ├── pdfer_theme.py
│   │   │   ├── pdfer_typography.py
│   │   │   └── STYLES.md
│   │   ├── playwrighter/
│   │   │   ├── config.yaml
│   │   │   └── playwrighter.py
│   │   ├── pptxer/
│   │   │   ├── config.yaml
│   │   │   ├── pptxer.py
│   │   │   ├── pptxer_audit.py
│   │   │   ├── pptxer_build.py
│   │   │   ├── pptxer_color.py
│   │   │   ├── pptxer_diagram.py
│   │   │   ├── pptxer_docmodel.py
│   │   │   ├── pptxer_draw.py
│   │   │   ├── pptxer_fonts.py
│   │   │   ├── pptxer_layout.py
│   │   │   ├── pptxer_media.py
│   │   │   ├── pptxer_nuance.py
│   │   │   ├── pptxer_render.py
│   │   │   ├── pptxer_theme.py
│   │   │   └── STYLES.md
│   │   ├── prompter/
│   │   │   ├── config.yaml
│   │   │   └── prompter.py
│   │   ├── pser/
│   │   │   ├── config.yaml
│   │   │   └── pser.py
│   │   ├── pythonxer/
│   │   │   ├── config.yaml
│   │   │   └── pythonxer.py
```

```
│   │   ├── raiser/
│   │   │   ├── config.yaml
│   │   │   └── raiser.py
│   │   ├── recmailer/
│   │   │   ├── config.yaml
│   │   │   └── recmailer.py
│   │   ├── recorder/
│   │   │   ├── config.yaml
│   │   │   └── recorder.py
│   │   ├── reviewer/
│   │   │   ├── config.yaml
│   │   │   └── reviewer.py
│   │   ├── scper/
│   │   │   ├── config.yaml
│   │   │   └── scper.py
│   │   ├── shoter/
│   │   │   ├── config.yaml
│   │   │   └── shoter.py
│   │   ├── sleeper/
│   │   │   ├── config.yaml
│   │   │   └── sleeper.py
│   │   ├── sqler/
│   │   │   ├── config.yaml
│   │   │   └── sqler.py
│   │   ├── ssher/
│   │   │   ├── config.yaml
│   │   │   └── ssher.py
│   │   ├── starter/
│   │   │   ├── config.yaml
│   │   │   └── starter.py
│   │   ├── stm32er/
│   │   │   ├── config.yaml
│   │   │   └── stm32er.py
│   │   ├── stopper/
│   │   │   ├── config.yaml
│   │   │   └── stopper.py
│   │   ├── summarizer/
│   │   │   ├── config.yaml
│   │   │   └── summarizer.py
│   │   ├── talker/
│   │   │   ├── config.yaml
│   │   │   └── talker.py
│   │   ├── telegrammer/
│   │   │   ├── config.yaml
│   │   │   ├── HOW_TO_GET_YOUR_TELEGRAM_ASSETS.md
│   │   │   ├── telegrammer.py
│   │   │   └── guide_assets/
│   │   │       ├── telegram_01_botfather_search.jpg
│   │   │       ├── telegram_02_newbot_token.jpg
│   │   │       ├── telegram_03_start_bot_getupdates.jpg
│   │   │       └── telegram_04_tlamatini_contacts.jpg
│   │   ├── teletlamatini/
│   │   │   ├── config.yaml
│   │   │   └── teletlamatini.py
│   │   ├── terminater/
│   │   │   ├── config.yaml
│   │   │   ├── terminater.py
│   │   │   ├── terminater_conpty.py
│   │   │   ├── terminater_engine.py
│   │   │   └── terminater_text.py
│   │   ├── torrenter/
│   │   │   ├── activity.py
│   │   │   ├── client.py
│   │   │   ├── config.yaml
│   │   │   ├── contract.py
│   │   │   ├── sources.py
│   │   │   └── torrenter.py
│   │   ├── unrealer/
│   │   │   ├── config.yaml
│   │   │   └── unrealer.py
│   │   ├── video_analyzer/
│   │   │   ├── config.yaml
│   │   │   ├── README.md
│   │   │   ├── video_analyzer.py
│   │   │   ├── video_content.py
│   │   │   └── video_fetch.py
│   │   ├── videoplayer/
│   │   │   ├── config.yaml
│   │   │   └── videoplayer.py
│   │   ├── whatsapper/
│   │   │   ├── config.yaml
│   │   │   ├── HOW_TO_GET_YOUR_WHATSAPP_ASSETS.md
│   │   │   ├── whatsapper.py
│   │   │   └── guide_assets/
│   │   │       ├── whatsapp_01_meta_create_app.jpg
│   │   │       ├── whatsapp_02_api_setup_keys.jpg
│   │   │       ├── whatsapp_03_test_recipient_send.jpg
│   │   │       ├── whatsapp_04_system_user_token.jpg
│   │   │       └── whatsapp_05_tlamatini_contacts.jpg
│   │   ├── whisperer/
│   │   │   ├── chat_worker.py
│   │   │   ├── config.yaml
│   │   │   ├── README.md
│   │   │   └── whisperer.py
│   │   └── windower/
│   │       ├── config.yaml
│   │       └── windower.py
│   ├── doc_generation/
│   │   ├── complete_project_docs.py
│   │   ├── dossier_content.py
│   │   ├── dossier_editor_content.py
│   │   ├── dossier_pdf.py
│   │   ├── dossier_pptx.py
│   │   ├── dossier_theme.py
│   │   ├── dossier_verify.py
│   │   ├── mardown_to_pdf.py
│   │   ├── refresh_project_docs.py
│   │   └── test_output.pdf
│   ├── images/
│   │   ├── ACPXKeysConfigureWizard.jpg
│   │   ├── ConfigureModels.jpg
│   │   ├── MenuConfig.jpg
│   │   ├── mockup.jpg
│   │   └── XAIHT-Tlamatini.mp4
│   ├── imaging/
│   │   ├── converter.py
│   │   ├── image_interpreter.py
│   │   └── screenshot.py
│   ├── interactor/
│   │   ├── __init__.py
│   │   ├── consumer.py
│   │   ├── context.py
│   │   ├── editor.mjs
│   │   ├── files.py
│   │   ├── hub.py
│   │   ├── protocol.py
│   │   └── tool.py
│   ├── management/
│   │   ├── __init__.py
│   │   └── commands/
│   │       ├── acpx_lint.py
│   │       ├── check_agent_runtimes.py
│   │       ├── check_prompt_flow_panel.py
│   │       ├── check_terminater.py
│   │       ├── enroll_torrenter.py
│   │       └── startserver.py
│   ├── monitors/
│   │   └── monitor_sql.py
│   ├── opus_client/
│   │   ├── claude_opus_client.py
│   │   ├── examples.py
│   │   └── README.md
│   ├── rag/
│   │   ├── __init__.py
│   │   ├── binary_guard.py
│   │   ├── config.py
│   │   ├── factory.py
│   │   ├── interaction.py
│   │   ├── interface.py
│   │   ├── loaders.py
│   │   ├── prompts.py
│   │   ├── retrieval.py
│   │   ├── splitters.py
│   │   ├── utils.py
│   │   └── chains/
│   │       ├── base.py
│   │       ├── basic.py
│   │       ├── history_aware.py
│   │       └── unified.py
│   ├── services/
│   │   ├── __init__.py
│   │   ├── agent_contracts.py
│   │   ├── agent_paths.py
│   │   ├── filesystem.py
│   │   ├── flow_compiler.py
│   │   ├── flow_knowledge.py
│   │   ├── flow_spec.py
│   │   ├── prompt_flow_panel.py
│   │   └── response_parser.py
│   ├── skills/
│   │   ├── __init__.py
│   │   ├── frontmatter.py
│   │   ├── harness.py
│   │   ├── io_contract.py
│   │   ├── redaction.py
│   │   ├── registry.py
│   │   └── validation.py
│   ├── skills_pkg/
│   │   ├── _meta/
│   │   │   ├── lint.py
│   │   │   └── schema.json
│   │   ├── acp_router/
│   │   │   └── SKILL.md
│   │   ├── adding_external_mcp/
│   │   │   ├── SKILL.md
│   │   │   └── references/
│   │   │       ├── external_mcp_catalog_format.md
│   │   │       ├── llm_reflection_research.md
```

```
│   │   │   │   ├── transport_guide.md
│   │   │   │   └── troubleshooting.md
│   │   ├── code_review/
│   │   │   └── SKILL.md
│   │   ├── create_new_agent/
│   │   │   └── SKILL.md
│   │   ├── create_new_mcp/
│   │   │   └── SKILL.md
│   │   ├── flow_making/
│   │   │   ├── SKILL.md
│   │   │   ├── references/
│   │   │   │   └── flw_schema.md
│   │   │   └── scripts/
│   │   │       ├── make_flow.py
│   │   │       └── result_to_flw.py
│   │   ├── github/
│   │   │   └── SKILL.md
│   │   ├── gmail/
│   │   │   └── SKILL.md
│   │   ├── hello_world/
│   │   │   └── SKILL.md
│   │   ├── jira/
│   │   │   └── SKILL.md
│   │   ├── kali_pentest/
│   │   │   └── SKILL.md
│   │   ├── notion/
│   │   │   └── SKILL.md
│   │   ├── roblox_studio/
│   │   │   └── SKILL.md
│   │   ├── security_audit/
│   │   │   └── SKILL.md
│   │   ├── setup_new_acpx_key/
│   │   │   └── SKILL.md
│   │   ├── skill_creator/
│   │   │   ├── SKILL.md
│   │   │   └── scripts/
│   │   │       └── quick_validate.py
│   │   ├── slack/
│   │   │   └── SKILL.md
│   │   ├── summarize/
│   │   │   └── SKILL.md
│   │   ├── tlamatini_allowed_hosts_tighten/
│   │   │   └── SKILL.md
│   │   ├── tlamatini_csrf_exempt_audit/
│   │   │   └── SKILL.md
│   │   ├── tlamatini_exec_report_row_adder/
│   │   │   └── SKILL.md
│   │   ├── tlamatini_flow_from_objective/
│   │   │   └── SKILL.md
│   │   ├── tlamatini_flw_doctor/
│   │   │   └── SKILL.md
│   │   ├── tlamatini_new_acp_agent/
│   │   │   └── SKILL.md
│   │   ├── tlamatini_planner_trace_replay/
│   │   │   └── SKILL.md
│   │   ├── tlamatini_static_version_bumper/
│   │   │   └── SKILL.md
│   │   ├── todoist/
│   │   │   └── SKILL.md
│   │   ├── trello/
│   │   │   └── SKILL.md
│   │   └── weather/
│   │       └── SKILL.md
│   ├── static/
│   │   └── agent/
│   │       ├── css/
│   │       │   ├── access_keys_wizard.css
│   │       │   ├── admin_theme.css
│   │       │   ├── agent_page.css
│   │       │   ├── agentic_control_panel.css
│   │       │   ├── avatar.css
│   │       │   ├── canvas_editor.css
│   │       │   ├── chat_dictation.css
│   │       │   ├── contacts_dialog.css
│   │       │   ├── context_gauge.css
│   │       │   ├── dialog_theme.css
│   │       │   ├── entry_pages.css
│   │       │   ├── external_mcps_dialog.css
│   │       │   ├── flow_canvas.css
│   │       │   ├── interaction_settings.css
│   │       │   ├── login.css
│   │       │   ├── markdown_canvas.css
│   │       │   ├── model_brain_tuning.css
│   │       │   ├── model_capacity.css
│   │       │   ├── pdf_canvas.css
│   │       │   ├── pdf_context_progress.css
│   │       │   ├── prompt_flow_panel.css
│   │       │   ├── skills_dialog.css
│   │       │   ├── terminater_dock.css
│   │       │   ├── tools_dialog.css
│   │       │   ├── ui_shell.css
│   │       │   ├── ui_splitters.css
│   │       │   ├── ui_tokens.css
│   │       │   ├── usage_dashboard.css
│   │       │   └── welcome.css
│   │       ├── img/
│   │       │   ├── spinner.svg
│   │       │   ├── Tlamatini.ico
│   │       │   └── avatar/
│   │       │       ├── ec_mc.jpg
│   │       │       ├── ec_mo.jpg
│   │       │       ├── eo_mc.jpg
│   │       │       └── eo_mo.jpg
│   │       ├── interactor/
│   │       │   ├── bridge.js
│   │       │   └── interactor.css
│   │       ├── js/
│   │       │   ├── access_keys_wizard.js
│   │       │   ├── acp-agent-connectors.js
│   │       │   ├── acp-canvas-core.js
│   │       │   ├── acp-canvas-undo.js
│   │       │   ├── acp-connection-status.js
│   │       │   ├── acp-control-buttons.js
│   │       │   ├── acp-editor-tools.js
│   │       │   ├── acp-file-io.js
│   │       │   ├── acp-flow-snapshot.js
│   │       │   ├── acp-globals.js
│   │       │   ├── acp-layout.js
│   │       │   ├── acp-parametrizer-dialog.js
│   │       │   ├── acp-running-state.js
│   │       │   ├── acp-session.js
│   │       │   ├── acp-undo-manager.js
│   │       │   ├── acp-validate.js
│   │       │   ├── agent_page_canvas.js
│   │       │   ├── agent_page_chat.js
│   │       │   ├── agent_page_context.js
│   │       │   ├── agent_page_dialogs.js
│   │       │   ├── agent_page_editing.js
│   │       │   ├── agent_page_init.js
│   │       │   ├── agent_page_layout.js
│   │       │   ├── agent_page_markdown.js
│   │       │   ├── agent_page_pdf.js
│   │       │   ├── agent_page_state.js
│   │       │   ├── agent_page_ui.js
│   │       │   ├── agentic_control_panel.js
│   │       │   ├── avatar.js
│   │       │   ├── avatar_presence.js
│   │       │   ├── avatar_size.js
│   │       │   ├── canvas_item_dialog.js
│   │       │   ├── chat_dictation.js
│   │       │   ├── chat_image_paste.js
│   │       │   ├── chat_page_runtime_poller.js
│   │       │   ├── chat_table_contrast.js
│   │       │   ├── checkbox_bulk_toggle.js
│   │       │   ├── compact_costs.js
│   │       │   ├── contacts_dialog.js
│   │       │   ├── context_gauge.js
│   │       │   ├── contextual_menus.js
│   │       │   ├── dialog_policy.js
│   │       │   ├── external_mcps_dialog.js
│   │       │   ├── file_drop_router.js
│   │       │   ├── flow-canvas-interactions.js
│   │       │   ├── interaction_settings.js
│   │       │   ├── interaction_speech.js
│   │       │   ├── mic_settings.js
│   │       │   ├── model_brain_tuning.js
│   │       │   ├── model_capacity.js
│   │       │   ├── pdf_canvas_viewer.js
│   │       │   ├── pdf_context_progress.js
│   │       │   ├── prompt-flow-panel-model.js
│   │       │   ├── prompt-flow-panel.js
│   │       │   ├── release_notes_renderer.js
│   │       │   ├── self_modify_switch.js
│   │       │   ├── shared-runtime-dialogs.js
│   │       │   ├── skills_dialog.js
│   │       │   ├── terminater_dock.js
│   │       │   ├── tools_dialog.js
│   │       │   ├── ui_splitters.js
│   │       │   ├── usage_dashboard.js
│   │       │   ├── voice_runtime.js
│   │       │   └── welcome_enter_default.js
│   │       ├── pdf/
│   │       │   └── canvas.html
│   │       ├── sounds/
│   │       │   ├── hypervisor_alert.wav
│   │       │   └── notification.wav
│   │       ├── vendor/
│   │       │   ├── canvas-editor/
│   │       │   │   ├── editor.js
│   │       │   │   ├── LICENSES.txt
│   │       │   │   └── manifest.json
│   │       │   ├── frontend/
│   │       │   │   ├── manifest.json
│   │       │   │   ├── bootstrap/
│   │       │   │   │   ├── LICENSE
│   │       │   │   │   └── dist/
│   │       │   │   │       ├── css/
│   │       │   │   │       │   ├── bootstrap.min.css
```

```
│   │   │   │   │   │   └── bootstrap.min.css.map
│   │   │   │   │   └── js/
│   │   │   │   │       ├── bootstrap.bundle.min.js
│   │   │   │   │       └── bootstrap.bundle.min.js.ma
│   │   │   │   │           » p
│   │   │   │   ├── dompurify/
│   │   │   │   │   ├── LICENSE
│   │   │   │   │   └── dist/
│   │   │   │   │       ├── purify.min.js
│   │   │   │   │       └── purify.min.js.map
│   │   │   │   ├── highlightjs/
│   │   │   │   │   ├── highlight.min.js
│   │   │   │   │   ├── LICENSE
│   │   │   │   │   ├── languages/
│   │   │   │   │   │   ├── bash.min.js
│   │   │   │   │   │   ├── c.min.js
│   │   │   │   │   │   ├── cpp.min.js
│   │   │   │   │   │   ├── css.min.js
│   │   │   │   │   │   ├── ini.min.js
│   │   │   │   │   │   ├── java.min.js
│   │   │   │   │   │   ├── javascript.min.js
│   │   │   │   │   │   ├── kotlin.min.js
│   │   │   │   │   │   ├── python.min.js
│   │   │   │   │   │   ├── rust.min.js
│   │   │   │   │   │   ├── sql.min.js
│   │   │   │   │   │   ├── typescript.min.js
│   │   │   │   │   │   ├── xml.min.js
│   │   │   │   │   │   └── yaml.min.js
│   │   │   │   │   └── styles/
│   │   │   │   │       └── atom-one-dark.min.css
│   │   │   │   ├── jquery/
│   │   │   │   │   ├── LICENSE.txt
│   │   │   │   │   └── dist/
│   │   │   │   │       ├── jquery.min.js
│   │   │   │   │       └── jquery.min.map
│   │   │   │   ├── jquery-ui/
│   │   │   │   │   ├── jquery-ui.min.js
│   │   │   │   │   └── LICENSE.txt
│   │   │   │   ├── marked/
│   │   │   │   │   └── LICENSE
│   │   │   │   ├── nunito/
│   │   │   │   │   ├── 400.css
│   │   │   │   │   ├── 700.css
│   │   │   │   │   ├── LICENSE
│   │   │   │   │   └── files/
│   │   │   │   │       ├── nunito-cyrillic-400-normal.wof
│   │   │   │   │       │   » f
│   │   │   │   │       ├── nunito-cyrillic-400-normal.wof
│   │   │   │   │       │   » f2
│   │   │   │   │       ├── nunito-cyrillic-700-normal.wof
│   │   │   │   │       │   » f
│   │   │   │   │       ├── nunito-cyrillic-700-normal.wof
│   │   │   │   │       │   » f2
│   │   │   │   │       ├── nunito-cyrillic-ext-400-normal
│   │   │   │   │       │   » .woff
│   │   │   │   │       ├── nunito-cyrillic-ext-400-normal
│   │   │   │   │       │   » .woff2
│   │   │   │   │       ├── nunito-cyrillic-ext-700-normal
│   │   │   │   │       │   » .woff
│   │   │   │   │       ├── nunito-cyrillic-ext-700-normal
│   │   │   │   │       │   » .woff2
│   │   │   │   │       ├── nunito-latin-400-normal.woff
│   │   │   │   │       ├── nunito-latin-400-normal.woff2
│   │   │   │   │       ├── nunito-latin-700-normal.woff
│   │   │   │   │       ├── nunito-latin-700-normal.woff2
│   │   │   │   │       ├── nunito-latin-ext-400-normal.wo
│   │   │   │   │       │   » ff
│   │   │   │   │       ├── nunito-latin-ext-400-normal.wo
│   │   │   │   │       │   » ff2
│   │   │   │   │       ├── nunito-latin-ext-700-normal.wo
│   │   │   │   │       │   » ff
│   │   │   │   │       ├── nunito-latin-ext-700-normal.wo
│   │   │   │   │       │   » ff2
│   │   │   │   │       ├── nunito-vietnamese-400-normal.w
│   │   │   │   │       │   » off
│   │   │   │   │       ├── nunito-vietnamese-400-normal.w
│   │   │   │   │       │   » off2
│   │   │   │   │       ├── nunito-vietnamese-700-normal.w
│   │   │   │   │       │   » off
│   │   │   │   │       └── nunito-vietnamese-700-normal.w
│   │   │   │   │           » off2
│   │   │   │   ├── xterm/
│   │   │   │   │   ├── LICENSE
│   │   │   │   │   └── css/
│   │   │   │   │       └── xterm.css
│   │   │   │   ├── xterm-addon-fit/
│   │   │   │   │   └── LICENSE
│   │   │   │   └── xterm-addon-web-links/
│   │   │   │       └── LICENSE
│   │   │   └── pdfjs/
│   │   │       ├── LICENSE
│   │   │       ├── VERSION
│   │   │       ├── build/
│   │   │       │   ├── pdf.mjs
│   │   │       │   └── pdf.worker.mjs
│   │   │       └── cmaps/
│   │   │           ├── 78-EUC-H.bcmap
│   │   │           ├── 78-EUC-V.bcmap
│   │   │           ├── 78-H.bcmap
│   │   │           ├── 78-RKSJ-H.bcmap
│   │   │           ├── 78-RKSJ-V.bcmap
│   │   │           ├── 78-V.bcmap
│   │   │           ├── 78ms-RKSJ-H.bcmap
│   │   │           ├── 78ms-RKSJ-V.bcmap
│   │   │           ├── 83pv-RKSJ-H.bcmap
│   │   │           ├── 90ms-RKSJ-H.bcmap
│   │   │           ├── 90ms-RKSJ-V.bcmap
│   │   │           ├── 90msp-RKSJ-H.bcmap
│   │   │           ├── 90msp-RKSJ-V.bcmap
│   │   │           ├── 90pv-RKSJ-H.bcmap
│   │   │           ├── 90pv-RKSJ-V.bcmap
│   │   │           ├── Add-H.bcmap
│   │   │           ├── Add-RKSJ-H.bcmap
│   │   │           ├── Add-RKSJ-V.bcmap
│   │   │           ├── Add-V.bcmap
│   │   │           ├── Adobe-CNS1-0.bcmap
│   │   │           ├── Adobe-CNS1-1.bcmap
│   │   │           ├── Adobe-CNS1-2.bcmap
│   │   │           ├── Adobe-CNS1-3.bcmap
│   │   │           ├── Adobe-CNS1-4.bcmap
│   │   │           ├── Adobe-CNS1-5.bcmap
│   │   │           ├── Adobe-CNS1-6.bcmap
│   │   │           ├── Adobe-CNS1-UCS2.bcmap
│   │   │           ├── Adobe-GB1-0.bcmap
│   │   │           ├── Adobe-GB1-1.bcmap
│   │   │           ├── Adobe-GB1-2.bcmap
│   │   │           ├── Adobe-GB1-3.bcmap
│   │   │           ├── Adobe-GB1-4.bcmap
│   │   │           ├── Adobe-GB1-5.bcmap
│   │   │           ├── Adobe-GB1-UCS2.bcmap
│   │   │           ├── Adobe-Japan1-0.bcmap
│   │   │           ├── Adobe-Japan1-1.bcmap
│   │   │           ├── Adobe-Japan1-2.bcmap
│   │   │           ├── Adobe-Japan1-3.bcmap
│   │   │           ├── Adobe-Japan1-4.bcmap
│   │   │           ├── Adobe-Japan1-5.bcmap
│   │   │           ├── Adobe-Japan1-6.bcmap
│   │   │           ├── Adobe-Japan1-UCS2.bcmap
│   │   │           ├── Adobe-Korea1-0.bcmap
│   │   │           ├── Adobe-Korea1-1.bcmap
│   │   │           ├── Adobe-Korea1-2.bcmap
│   │   │           ├── Adobe-Korea1-UCS2.bcmap
│   │   │           ├── B5-H.bcmap
│   │   │           ├── B5-V.bcmap
│   │   │           ├── B5pc-H.bcmap
│   │   │           ├── B5pc-V.bcmap
│   │   │           ├── CNS-EUC-H.bcmap
│   │   │           ├── CNS-EUC-V.bcmap
│   │   │           ├── CNS1-H.bcmap
│   │   │           ├── CNS1-V.bcmap
│   │   │           ├── CNS2-H.bcmap
│   │   │           ├── CNS2-V.bcmap
│   │   │           ├── ETen-B5-H.bcmap
│   │   │           ├── ETen-B5-V.bcmap
│   │   │           ├── ETenms-B5-H.bcmap
│   │   │           ├── ETenms-B5-V.bcmap
│   │   │           ├── ETHK-B5-H.bcmap
│   │   │           ├── ETHK-B5-V.bcmap
│   │   │           ├── EUC-H.bcmap
│   │   │           ├── EUC-V.bcmap
│   │   │           ├── Ext-H.bcmap
│   │   │           ├── Ext-RKSJ-H.bcmap
│   │   │           ├── Ext-RKSJ-V.bcmap
│   │   │           ├── Ext-V.bcmap
│   │   │           ├── GB-EUC-H.bcmap
│   │   │           ├── GB-EUC-V.bcmap
│   │   │           ├── GB-H.bcmap
│   │   │           ├── GB-V.bcmap
│   │   │           ├── GBK-EUC-H.bcmap
│   │   │           ├── GBK-EUC-V.bcmap
│   │   │           ├── GBK2K-H.bcmap
│   │   │           ├── GBK2K-V.bcmap
│   │   │           ├── GBKp-EUC-H.bcmap
│   │   │           ├── GBKp-EUC-V.bcmap
│   │   │           ├── GBpc-EUC-H.bcmap
│   │   │           ├── GBpc-EUC-V.bcmap
│   │   │           ├── GBT-EUC-H.bcmap
│   │   │           ├── GBT-EUC-V.bcmap
│   │   │           ├── GBT-H.bcmap
│   │   │           ├── GBT-V.bcmap
│   │   │           ├── GBTpc-EUC-H.bcmap
│   │   │           ├── GBTpc-EUC-V.bcmap
│   │   │           ├── H.bcmap
│   │   │           ├── Hankaku.bcmap
│   │   │           ├── Hiragana.bcmap
│   │   │           ├── HKdla-B5-H.bcmap
│   │   │           ├── HKdla-B5-V.bcmap
│   │   │           ├── HKdlb-B5-H.bcmap
```

```
        │   │   │   │   ├── HKdlb-B5-V.bcmap
        │   │   │   │   ├── HKgccs-B5-H.bcmap
        │   │   │   │   ├── HKgccs-B5-V.bcmap
        │   │   │   │   ├── HKm314-B5-H.bcmap
        │   │   │   │   ├── HKm314-B5-V.bcmap
        │   │   │   │   ├── HKm471-B5-H.bcmap
        │   │   │   │   ├── HKm471-B5-V.bcmap
        │   │   │   │   ├── HKscs-B5-H.bcmap
        │   │   │   │   ├── HKscs-B5-V.bcmap
        │   │   │   │   ├── Katakana.bcmap
        │   │   │   │   ├── KSC-EUC-H.bcmap
        │   │   │   │   ├── KSC-EUC-V.bcmap
        │   │   │   │   ├── KSC-H.bcmap
        │   │   │   │   ├── KSC-Johab-H.bcmap
        │   │   │   │   ├── KSC-Johab-V.bcmap
        │   │   │   │   ├── KSC-V.bcmap
        │   │   │   │   ├── KSCms-UHC-H.bcmap
        │   │   │   │   ├── KSCms-UHC-HW-H.bcmap
        │   │   │   │   ├── KSCms-UHC-HW-V.bcmap
        │   │   │   │   ├── KSCms-UHC-V.bcmap
        │   │   │   │   ├── KSCpc-EUC-H.bcmap
        │   │   │   │   ├── KSCpc-EUC-V.bcmap
        │   │   │   │   ├── LICENSE
        │   │   │   │   ├── NWP-H.bcmap
        │   │   │   │   ├── NWP-V.bcmap
        │   │   │   │   ├── RKSJ-H.bcmap
        │   │   │   │   ├── RKSJ-V.bcmap
        │   │   │   │   ├── Roman.bcmap
        │   │   │   │   ├── UniCNS-UCS2-H.bcmap
        │   │   │   │   ├── UniCNS-UCS2-V.bcmap
        │   │   │   │   ├── UniCNS-UTF16-H.bcmap
        │   │   │   │   ├── UniCNS-UTF16-V.bcmap
        │   │   │   │   ├── UniCNS-UTF32-H.bcmap
        │   │   │   │   ├── UniCNS-UTF32-V.bcmap
        │   │   │   │   ├── UniCNS-UTF8-H.bcmap
        │   │   │   │   ├── UniCNS-UTF8-V.bcmap
        │   │   │   │   ├── UniGB-UCS2-H.bcmap
        │   │   │   │   ├── UniGB-UCS2-V.bcmap
        │   │   │   │   ├── UniGB-UTF16-H.bcmap
        │   │   │   │   ├── UniGB-UTF16-V.bcmap
        │   │   │   │   ├── UniGB-UTF32-H.bcmap
        │   │   │   │   ├── UniGB-UTF32-V.bcmap
        │   │   │   │   ├── UniGB-UTF8-H.bcmap
        │   │   │   │   ├── UniGB-UTF8-V.bcmap
        │   │   │   │   ├── UniJIS-UCS2-H.bcmap
        │   │   │   │   ├── UniJIS-UCS2-HW-H.bcmap
        │   │   │   │   ├── UniJIS-UCS2-HW-V.bcmap
        │   │   │   │   ├── UniJIS-UCS2-V.bcmap
        │   │   │   │   ├── UniJIS-UTF16-H.bcmap
        │   │   │   │   ├── UniJIS-UTF16-V.bcmap
        │   │   │   │   ├── UniJIS-UTF32-H.bcmap
        │   │   │   │   ├── UniJIS-UTF32-V.bcmap
        │   │   │   │   ├── UniJIS-UTF8-H.bcmap
        │   │   │   │   ├── UniJIS-UTF8-V.bcmap
        │   │   │   │   ├── UniJIS2004-UTF16-H.bcmap
        │   │   │   │   ├── UniJIS2004-UTF16-V.bcmap
        │   │   │   │   ├── UniJIS2004-UTF32-H.bcmap
        │   │   │   │   ├── UniJIS2004-UTF32-V.bcmap
        │   │   │   │   ├── UniJIS2004-UTF8-H.bcmap
        │   │   │   │   ├── UniJIS2004-UTF8-V.bcmap
        │   │   │   │   ├── UniJISPro-UCS2-HW-V.bcmap
        │   │   │   │   ├── UniJISPro-UCS2-V.bcmap
        │   │   │   │   ├── UniJISPro-UTF8-V.bcmap
        │   │   │   │   ├── UniJISX0213-UTF32-H.bcmap
        │   │   │   │   ├── UniJISX0213-UTF32-V.bcmap
        │   │   │   │   ├── UniJISX02132004-UTF32-H.bcmap
        │   │   │   │   ├── UniJISX02132004-UTF32-V.bcmap
        │   │   │   │   ├── UniKS-UCS2-H.bcmap
        │   │   │   │   ├── UniKS-UCS2-V.bcmap
        │   │   │   │   ├── UniKS-UTF16-H.bcmap
        │   │   │   │   ├── UniKS-UTF16-V.bcmap
        │   │   │   │   ├── UniKS-UTF32-H.bcmap
        │   │   │   │   ├── UniKS-UTF32-V.bcmap
        │   │   │   │   ├── UniKS-UTF8-H.bcmap
        │   │   │   │   ├── UniKS-UTF8-V.bcmap
        │   │   │   │   ├── V.bcmap
        │   │   │   │   └── WP-Symbol.bcmap
        │   │   │   ├── iccs/
        │   │   │   │   ├── CGATS001Compat-v2-micro.icc
        │   │   │   │   └── LICENSE
        │   │   │   ├── standard_fonts/
        │   │   │   │   ├── FoxitDingbats.pfb
        │   │   │   │   ├── FoxitFixed.pfb
        │   │   │   │   ├── FoxitFixedBold.pfb
        │   │   │   │   ├── FoxitFixedBoldItalic.pfb
        │   │   │   │   ├── FoxitFixedItalic.pfb
        │   │   │   │   ├── FoxitSerif.pfb
        │   │   │   │   ├── FoxitSerifBold.pfb
        │   │   │   │   ├── FoxitSerifBoldItalic.pfb
        │   │   │   │   ├── FoxitSerifItalic.pfb
        │   │   │   │   ├── FoxitSymbol.pfb
        │   │   │   │   ├── LiberationSans-Bold.ttf
        │   │   │   │   ├── LiberationSans-BoldItalic.ttf
        │   │   │   │   ├── LiberationSans-Italic.ttf
        │   │   │   │   ├── LiberationSans-Regular.ttf
        │   │   │   │   ├── LICENSE_FOXIT
        │   │   │   │   └── LICENSE_LIBERATION
        │   │   │   ├── wasm/
        │   │   │   │   ├── jbig2.wasm
        │   │   │   │   ├── jbig2_nowasm_fallback.js
        │   │   │   │   ├── LICENSE_JBIG2
        │   │   │   │   ├── LICENSE_OPENJPEG
        │   │   │   │   ├── LICENSE_PDFJS_JBIG2
        │   │   │   │   ├── LICENSE_PDFJS_OPENJPEG
        │   │   │   │   ├── LICENSE_PDFJS_QCMS
        │   │   │   │   ├── LICENSE_QCMS
        │   │   │   │   ├── openjpeg.wasm
        │   │   │   │   ├── openjpeg_nowasm_fallback.js
        │   │   │   │   ├── qcms_bg.wasm
        │   │   │   │   ├── quickjs-eval.js
        │   │   │   │   └── quickjs-eval.wasm
        │   │   │   └── web/
        │   │   │       ├── pdf_viewer.css
        │   │   │       ├── pdf_viewer.mjs
        │   │   │       └── images/
        │   │   │           ├── altText_add.svg
        │   │   │           ├── altText_disclaimer.svg
        │   │   │           ├── altText_done.svg
        │   │   │           ├── altText_spinner.svg
        │   │   │           ├── altText_warning.svg
        │   │   │           ├── annotation-check.svg
        │   │   │           ├── annotation-comment.svg
        │   │   │           ├── annotation-help.svg
        │   │   │           ├── annotation-insert.svg
        │   │   │           ├── annotation-key.svg
        │   │   │           ├── annotation-newparagraph.svg
        │   │   │           ├── annotation-noicon.svg
        │   │   │           ├── annotation-note.svg
        │   │   │           ├── annotation-paperclip.svg
        │   │   │           ├── annotation-paragraph.svg
        │   │   │           ├── annotation-pushpin.svg
        │   │   │           ├── checkmark.svg
        │   │   │           ├── comment-actionsButton.svg
        │   │   │           ├── comment-closeButton.svg
        │   │   │           ├── comment-editButton.svg
        │   │   │           ├── comment-popup-editButton.svg
        │   │   │           ├── cursor-editorFreeHighlight.svg
        │   │   │           ├── cursor-editorFreeText.svg
        │   │   │           ├── cursor-editorInk.svg
        │   │   │           ├── cursor-editorTextHighlight.svg
        │   │   │           ├── editor-toolbar-delete.svg
        │   │   │           ├── editor-toolbar-edit.svg
        │   │   │           ├── findbarButton-next.svg
        │   │   │           ├── findbarButton-previous.svg
        │   │   │           ├── gv-toolbarButton-download.svg
        │   │   │           ├── loading-icon.gif
        │   │   │           ├── loading.svg
        │   │   │           ├── messageBar_closingButton.svg
        │   │   │           ├── messageBar_info.svg
        │   │   │           ├── messageBar_warning.svg
        │   │   │           ├── pages_closeButton.svg
        │   │   │           ├── pages_selected.svg
        │   │   │           ├── pages_viewArrow.svg
        │   │   │           ├── pages_viewButton.svg
        │   │   │           ├── secondaryToolbarButton-documen
        │   │   │           │   » tProperties.svg
        │   │   │           ├── secondaryToolbarButton-firstPa
        │   │   │           │   » ge.svg
        │   │   │           ├── secondaryToolbarButton-handToo
        │   │   │           │   » l.svg
        │   │   │           ├── secondaryToolbarButton-lastPag
        │   │   │           │   » e.svg
        │   │   │           ├── secondaryToolbarButton-rotateC
        │   │   │           │   » cw.svg
        │   │   │           ├── secondaryToolbarButton-rotateC
        │   │   │           │   » w.svg
        │   │   │           ├── secondaryToolbarButton-scrollH
        │   │   │           │   » orizontal.svg
        │   │   │           ├── secondaryToolbarButton-scrollP
        │   │   │           │   » age.svg
        │   │   │           ├── secondaryToolbarButton-scrollV
        │   │   │           │   » ertical.svg
        │   │   │           ├── secondaryToolbarButton-scrollW
        │   │   │           │   » rapped.svg
        │   │   │           ├── secondaryToolbarButton-selectT
        │   │   │           │   » ool.svg
        │   │   │           ├── secondaryToolbarButton-spreadE
        │   │   │           │   » ven.svg
        │   │   │           ├── secondaryToolbarButton-spreadN
        │   │   │           │   » one.svg
        │   │   │           ├── secondaryToolbarButton-spreadO
        │   │   │           │   » dd.svg
        │   │   │           ├── signature-properties-row-check
        │   │   │           │   » .svg
        │   │   │           ├── toolbarButton-bookmark.svg
        │   │   │           ├── toolbarButton-currentOutlineIt
        │   │   │           │   » em.svg
        │   │   │           └── toolbarButton-download.svg
```

```
│   │   │   │   ├── toolbarButton-editorFreeText.s
│   │   │   │   │   » vg
│   │   │   │   ├── toolbarButton-editorHighlight.
│   │   │   │   │   » svg
│   │   │   │   ├── toolbarButton-editorInk.svg
│   │   │   │   ├── toolbarButton-editorSignature.
│   │   │   │   │   » svg
│   │   │   │   ├── toolbarButton-editorStamp.svg
│   │   │   │   ├── toolbarButton-menuArrow.svg
│   │   │   │   ├── toolbarButton-menuArrowNova.sv
│   │   │   │   │   » g
│   │   │   │   ├── toolbarButton-openFile.svg
│   │   │   │   ├── toolbarButton-pageDown.svg
│   │   │   │   ├── toolbarButton-pageUp.svg
│   │   │   │   ├── toolbarButton-presentationMode
│   │   │   │   │   » .svg
│   │   │   │   ├── toolbarButton-print.svg
│   │   │   │   ├── toolbarButton-search.svg
│   │   │   │   ├── toolbarButton-secondaryToolbar
│   │   │   │   │   » Toggle.svg
│   │   │   │   ├── toolbarButton-signaturePropert
│   │   │   │   │   » iesError.svg
│   │   │   │   ├── toolbarButton-signaturePropert
│   │   │   │   │   » iesVerified.svg
│   │   │   │   ├── toolbarButton-viewAttachments.
│   │   │   │   │   » svg
│   │   │   │   ├── toolbarButton-viewLayers.svg
│   │   │   │   ├── toolbarButton-viewOutline.svg
│   │   │   │   ├── toolbarButton-viewsManagerTogg
│   │   │   │   │   » le.svg
│   │   │   │   ├── toolbarButton-viewThumbnail.sv
│   │   │   │   │   » g
│   │   │   │   ├── toolbarButton-zoomIn.svg
│   │   │   │   ├── toolbarButton-zoomOut.svg
│   │   │   │   ├── treeitem-collapsed.svg
│   │   │   │   └── treeitem-expanded.svg
│   │   │   └── video/
│   │   │       └── XAIHT-Tlamatini.mp4
│   │   ├── templates/
│   │   │   ├── admin/
│   │   │   │   └── base_site.html
│   │   │   └── agent/
│   │   │       ├── agent_page.html
│   │   │       ├── agentic_control_panel.html
│   │   │       ├── interaction_settings.html
│   │   │       ├── interactor.html
│   │   │       ├── login.html
│   │   │       ├── prompt_flow_panel.html
│   │   │       ├── torrenter_sources.html
│   │   │       ├── usage_dialog.html
│   │   │       └── welcome.html
│   │   ├── test_fixtures/
│   │   │   └── talker_orpheus_tokens.txt
│   │   ├── TlamatiniSourceCode/
│   │   │   └── README.md
│   │   └── torrenter_workspace/
│   │       ├── layout.js
│   │       ├── workspace.css
│   │       ├── workspace.html
│   │       └── workspace.js
│   ├── DB/
│   │   ├── Older/
│   │   │   └── README.md
│   │   └── ToLoad/
│   │       └── README.md
│   ├── jd-cli/
│   │   ├── jd-cli.bat
│   │   └── jd-cli.jar
│   ├── tests_e2e/
│   │   ├── oob_child_helper.py
│   │   ├── README.md
│   │   ├── test_create_flow_visual.py
│   │   ├── test_ctrl_c_quits_visible.py
│   │   ├── test_db_backup_set_visible.py
│   │   ├── test_oob_browser_visible.py
│   │   ├── test_oob_namu_unit.py
│   │   ├── test_prompts_catalog_visual.py
│   │   ├── test_self_healing_visual.py
│   │   ├── test_voice_visual.py
│   │   ├── Tlamatini-Whatsapper-Daily-Tests.py
│   │   ├── torrenter_books.py
│   │   ├── torrenter_console_windows.py
│   │   ├── torrenter_live.py
│   │   └── torrenter_panel.py
│   └── tlamatini/
│       ├── __init__.py
│       ├── asgi.py
│       ├── context_processors.py
│       ├── logging_filters.py
│       ├── middleware.py
│       ├── settings.py
│       ├── urls.py
│       └── wsgi.py
```



Tlamatini

one who knows

Conceived, designed and built by **Angela López Mendoza**. Every fact in this dossier was counted from the source tree when it was generated; regenerate it at any time with `refresh_project_docs.py` and it will tell the truth again.

Set in Palatino Linotype, Segoe UI and Consolas. Generated November 8, 2026 at 02:06 (UTC-06:00) from commit 52e97879. MIT license · github.com/XAIHT/Tlamatini