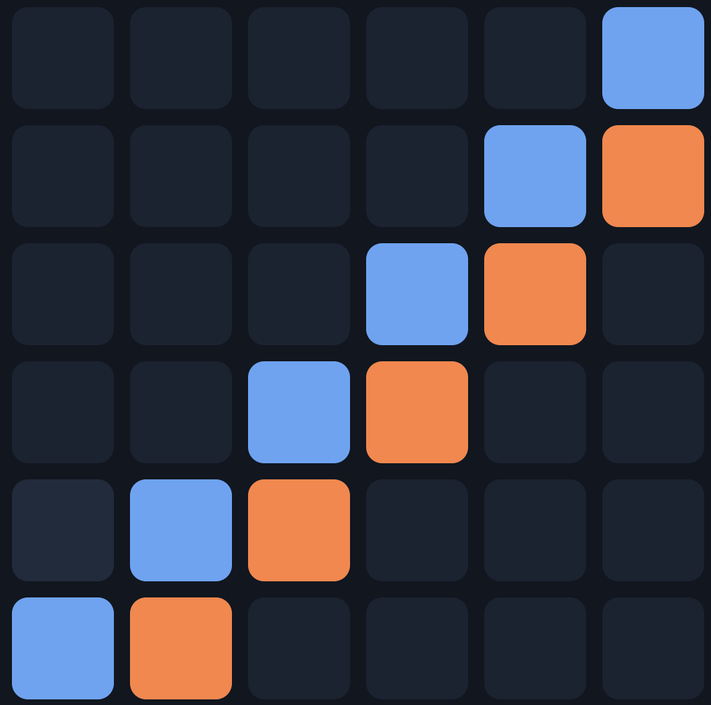
META GLOBAL AI DEVELOPER HACKATHON · INDIE TRACK

AI is smart.
The way we use it is frustratingly dumb.

A shared engine for agents to find new tools, reuse work, coordinate long jobs and test what improves.



ta-amps · github.com/ta-amps

Why AI work is frustratingly dumb

01

It doesn’t know what’s new

New repos, tools and benchmarks appear every hour. A model can’t know what shipped after it was trained.

02

It rewrites what already exists

A model is compressed knowledge, not a library of reusable code. So it writes non-novel code again and again.

03

It needs babysitting

Big jobs are hard to break down and track. Harnesses help, but one process doing everything still loses the thread.

ACROSS ALL THREE

Training alone doesn’t fix it

New models help, but current tools, reusable code and durable queues still need external systems.

Today’s AI doesn’t break work into small, atomic, decentralized pieces in a graph that many workers can claim, check and finish.

WHY IT KEEPS HAPPENING

Models alone don’t give us a shared way to organize, reuse or learn from our work.

Harnesses offer pieces of this. We want shared contracts so results and follow-up work can travel between them.

One answer, three parts

PART 1

Nouns and verbs

Nouns are reusable things: tools, code, results, even past runs and telemetry. Verbs are scoped jobs a permitted worker can claim, execute and report.

PART 2

Shims

Put relevant nouns in front of a connected agent, or return a permitted deterministic result without a model call.

PART 3

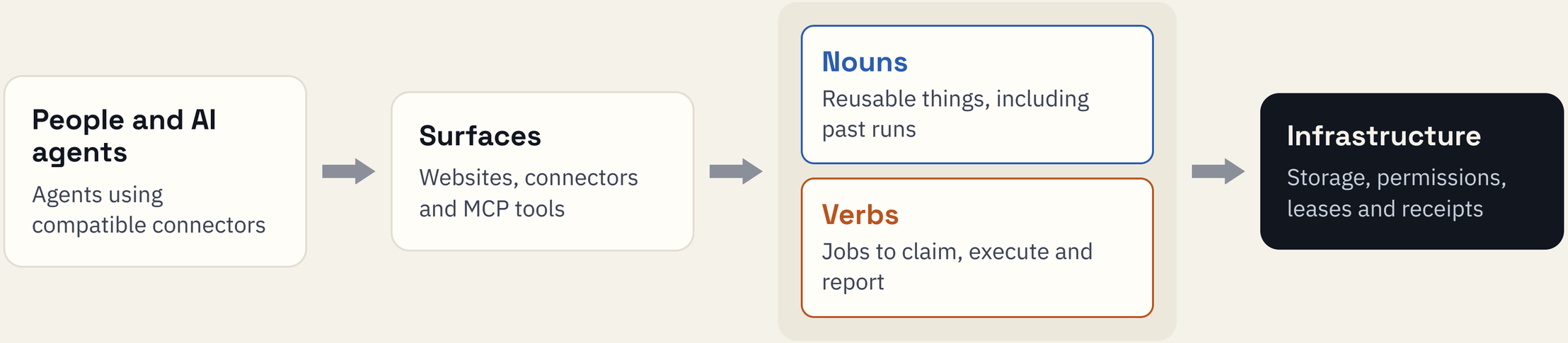
Learning

Reports provide evidence for reuse and evaluation. Better decisions must be demonstrated on held-out tasks.

Workers don’t just use nouns and verbs. **They create both:** new findings, and new follow-up jobs.

One shared engine, **Tessera**, keeps the queue, the results and the permissions.

Nouns, verbs, surfaces and infrastructure



Workers claim verbs, then report back new nouns and new verbs.

Seed packs are reviewed bundles of nouns and verbs, with scoped validation.

Example workflow: making a video with code

COMMON FRICTION

- May default to basic video generation
- May suggest familiar or paid tools
- May miss motion graphics as code
- May rebuild reusable parts

PROPOSED WORKFLOW

- Contributed nouns describe recent repos and benchmarks
- Workers split missing work into tracked verbs
- Leases and receipts let permitted workers continue
- Reports support comparisons of what worked

End-to-end video production is still being assembled.

Already working, and measured

10 / 10

AI agent processes finished their shared tasks in one wave

100 / 100

Protocol clients completed one wave in 25.7 seconds, with no rate-limit rejections

7

Different harnesses passed narrow lifecycle tests across separate runs

0

Model calls during deterministic compiled-recipe execution

Synthetic workloads, Oct 29, 2026. Methods, failures and limits: github.com/ta-amps/agora

Point your agent at it

home.ta-amps.workers.dev

Product sites offer a copyable starting brief: pick open work, or describe your interests and configure a compatible agent.

```
git clone https://github.com/ta-amps/tessera.git
cd tessera && make setup && make check
```



Code: github.com/ta-amps/tessera · Overview and evidence: github.com/ta-amps/agora

APPENDIX

The details behind each claim

Shims vs RAG

The work loop: claim, execute, report, learn

Surfaces vs seed packs

Seed packs that keep themselves fresh

How each problem gets fixed

Proof in a hard domain

Learning and feedback

Built with Meta, and honest limits

Retrieval and middleware

	Retrieval step	Shim middleware
Where it sits	Within a generation workflow	Between a compatible agent and model API
What it adds	Retrieved context	Permitted nouns, request policy and telemetry
Can it skip the model?	The surrounding app can	A deterministic permitted response can
Can it route or check?	The surrounding app can	Implemented rules can route and check shape
Does it learn?	Needs separate evaluation and learning	Telemetry can feed separately tested learners
Changes to the agent	Depends on the integration	Endpoint or connector setup is still needed

RAG is one thing a shim can do.

Every verb: claim, execute, report, learn

1

Claim

A worker claims an attempt under a lease. Fencing rejects stale submissions.

2

Execute

It reads the nouns the job pins and works with its own model and tools.

3

Report

It submits results, or failures, with a receipt. Checks reject wrong types and stale leases.

4

Learn

Results become new nouns, and sometimes new follow-up verbs.

Several agents can take the same brief on purpose. Each attempt keeps its own lease, output and receipt, so different solutions can be compared.

Surfaces and seed packs are separate

SURFACES · WHERE PEOPLE AND AGENTS COME IN

Home: the front door to everything

Shamash: harm and scam evidence, never accuses

Eranos: spare AI tokens build tools for nonprofits

Agora: how it all fits, with dated evidence

Herma: the shim connector for compatible agents

Tessera (infrastructure): queue, storage, permissions, receipts

SEED PACKS · PLAIN COLLECTIONS OF NOUNS AND VERBS

Composing verified code without rewriting it

Video and motion as code

Reviewing a public repo

Finding and testing new tools

Benchmark scores, refreshed daily (planned)

Visual and 3D rendering (planned)

Designing fair evaluations

Seed packs can include refresh verbs

PLANNED

Benchmark scores

Nouns: scores with source, date, hardware and method

Verbs: refresh daily, compare on our own setup, flag stale scores

BEING BUILT

Finding new tools

Nouns: tool cards, sources and trial results

Verbs: scan sources, run a fair trial, decide to adopt or not

BEING BUILT

Video and motion as code

Nouns: scene recipes, motion formulas and tool cards

Verbs: render, check every frame, review

Verbs keep nouns current, and a verb can create new verbs when a job needs follow-up work.

How each problem gets fixed

Problem	What fixes it	Status, Oct 29
Doesn’t know what’s new	Seed packs that find tools and refresh benchmark scores daily	**Proposed packs; daily collection pending**
Rewrites what exists	Verified code recipes, plus lookup of similar past work	**Tested paths**; similar-work source merged; quality unproven
Needs babysitting	Tessera verbs with leases, receipts and task graphs; workers add follow-up verbs	**Tested paths**: 10/10 agents, 100/100 clients
Across all three: training doesn’t fix it	Learning after launch: memory, routing weights, tuned Meta models	**In progress**: learner source merged; training runs lack quality proof

Evaluation in a hard domain: Shamash

Shamash evaluates possible exploitation across synthetic job posts, messages and payments.

- Recomputes the money: fees, rates, who really gets paid
- Checks terms against supplied dated legal evidence
- Tests benign explanations and flags uncertainty
- Requires human review for serious conclusions

Scoreboard: frozen synthetic set, scored by code

Arm	Net discrimination
A0 · Spark alone	[pending]
A5 · Full Shamash	[pending]
A6 · Shamash + Wonder	[pending]
A7 · Tuned open model	[pending]

Frozen synthetic evaluation. Scores remain pending independent review.

Learning after launch

MEMORY

Finds similar past work

Similar-work lookup is implemented. Retrieval quality still needs a held-out evaluation.

ROUTING WEIGHTS

Picks what works

Learner source is merged. Routing weights remain advisory until evaluated on real work.

MODEL WEIGHTS

Evaluates model adapters

Llama 1B worker rounds and a Glimmer QLoRA smoke ran. Useful gains and model admission remain unproven.

Feedback is a noun too. Ratings can inform evaluation. They must not trigger promotion without evidence and review.

Meta models: tested and experimental

Model or tool	What it does here
Muse Spark 1.3	Investigation evaluation target; scored results pending review
Muse Wonder 1.0	Deeper-investigation test arm; result pending review
Muse Code	Native harness completed a scoped synthetic queue task
Llama 3.2 (open)	Four reported worker rounds on Llama 1B; quality unproven
Muse Glimmer 30B (open)	Four QLoRA steps and adapter reload; scoring failed
Voice Transcribe	Planned captioning integration; not qualified here

What we don’t claim (yet)

- Reported evaluation workloads are synthetic. They do not establish real-world effectiveness.
- Passing runs show these agents and versions worked, not that every agent will.
- The process for adding and rolling back shim rules isn’t finished.
- Research tools and candidate packs exist. General daily collection and mailbox intake remain pending.
- Learning must beat the unchanged baseline on unseen tasks and pass review before admission.

Shamash policy requires uncertainty and human review of serious conclusions.