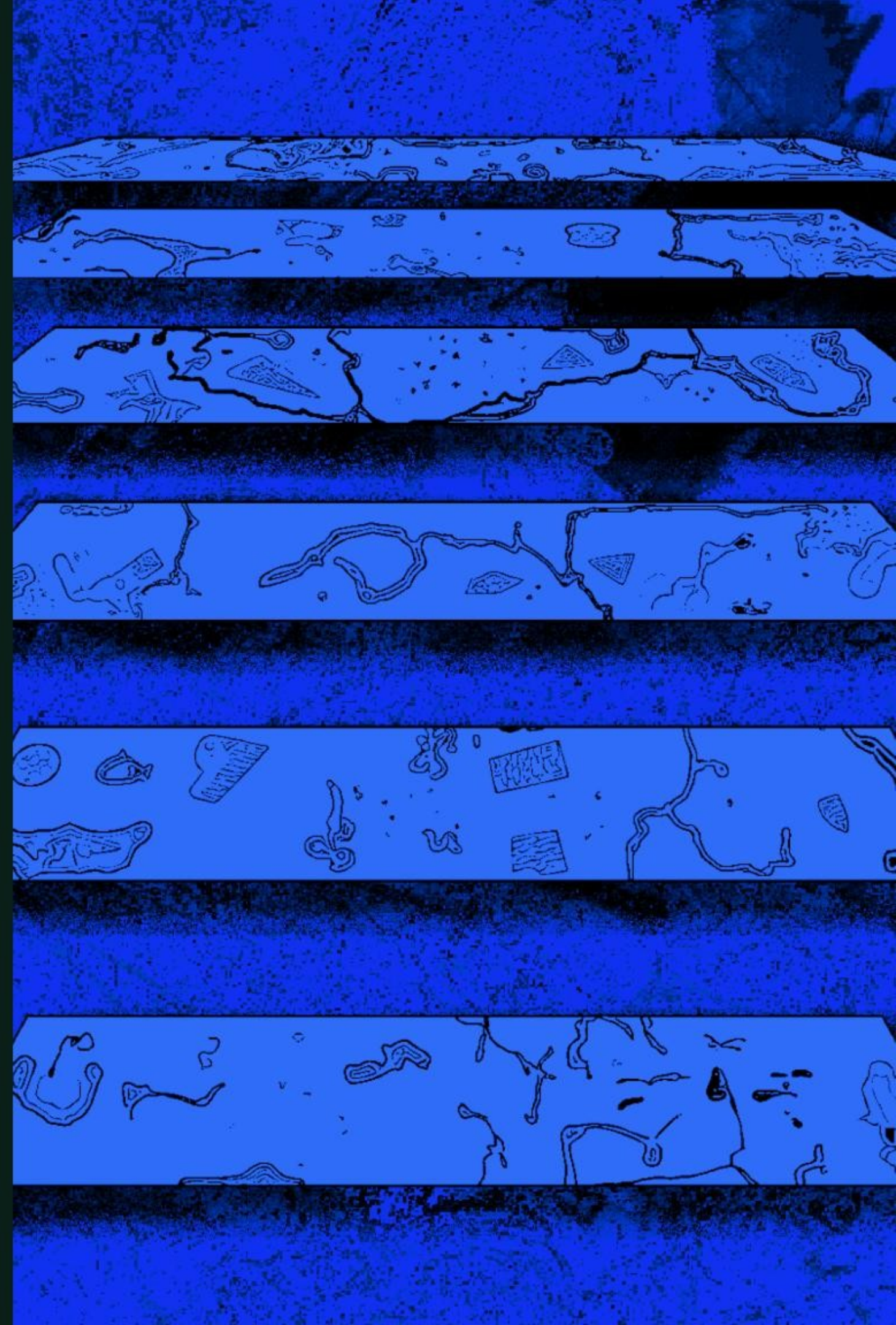


TWO-DAY HANDS-ON WORKSHOP

AI Kitcraft

Kit → Factory → Bridge

github.com/protocolvision/workshop-kitkraft · Discord #workshop-kitkraft



About the Protocol Institute

THE PROTOCOL INSTITUTE

Accelerating Order.

A research, education, media and scene-making organization for protocols — how coordination mechanisms get designed, adopted and maintained.

Publishes *Protocolized*, the magazine of New Nature · runs the SIGs, including Protocols for Business · hosts the Protocol Symposium, where this workshop is one of five running in parallel.

SACHIN BENNY

Protocol Institute · writes *Summer Lightning*

Wrote "LLMs Pre-Commodify Ideas" (June 2026), where precommodification comes from, and the AI-and-tractors piece behind the kit-phase argument in Session 1. His agent captures the Session 4 synthesis.

RAFA

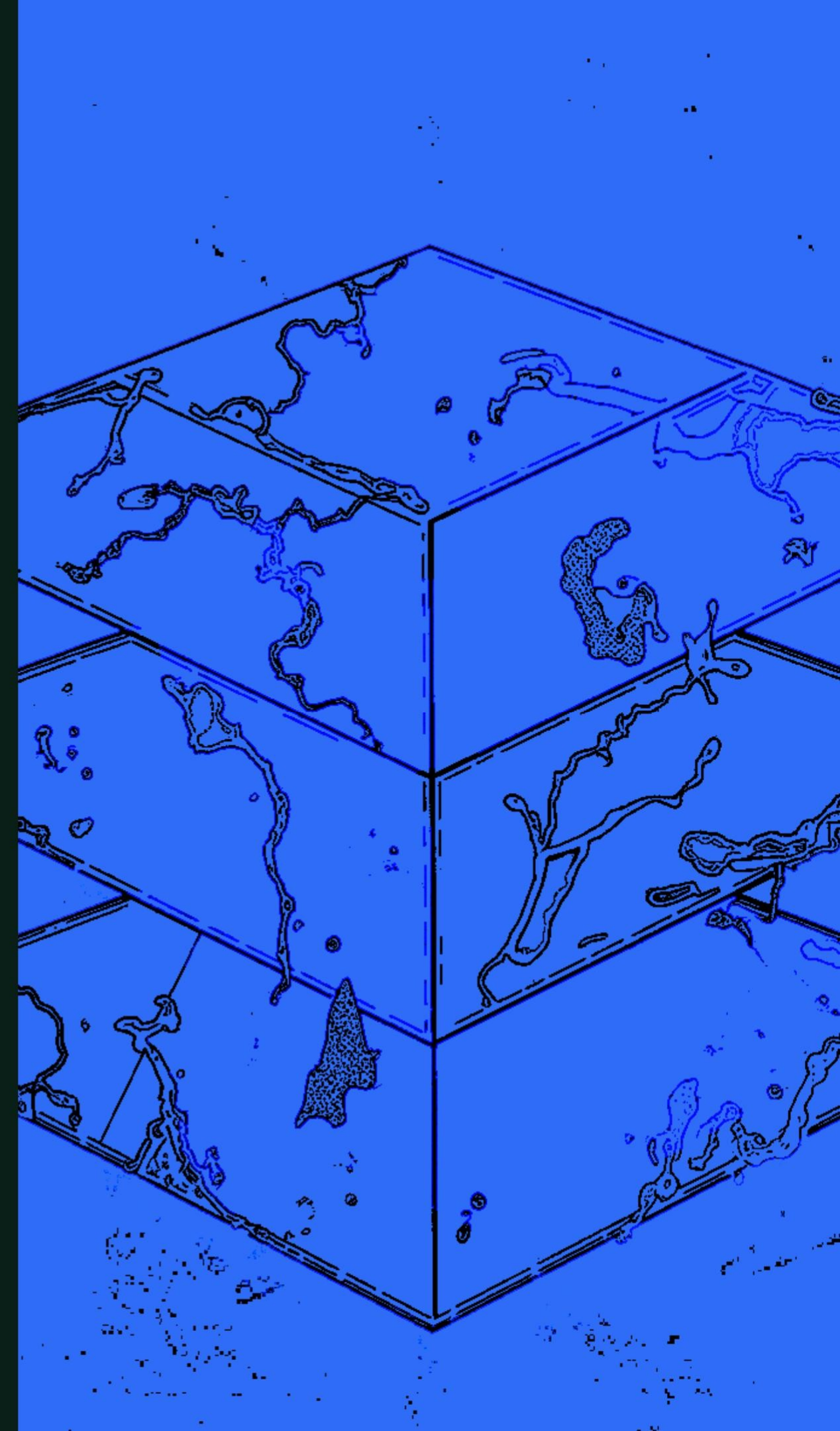
Protocols for Business SIG

Wrote "Hard Worlds for Big Organizations" (April 2026) and named the four quadrants we use as our lens. Runs the rate-schedule factory in `facilitator/rafa/` that is a bridge target in Session 3.

SESSION 1 • 60 MINUTES

Kits

Everyone uses AI their own way, and nobody else can see it. You leave with an inventory of what you already do.



Over two days, we will share how we work with AI and build bridges.

NOW · S1**Kits**

Where you are, what a kit is, your inventory

LATER 20:00**Factories**

Your kit runs without you

TUE 15:30**Bridges**

Your agent uses someone else's kit

TUE 20:00**Show and tell**

The slides are your files

One rule for everyone: save everything (public) in the workshop repo — github.com/protocolvision/workshop-kitkraft

Before we start... testing 1,2,3...

WHAT IS RECORDED

Voice in #kafka and text in #workshop-kitkraft, transcribed with names

Transcripts go into the public repo

Breakout rooms may be recorded

BUT...

Ask either of us to remove any line — no reason needed, any time

Say so now if you would rather be off the transcript

Nothing in a public kit you would not put on a public website

Keep sensitive information in your local folder on your own machine. The repo is public from the first commit.

Share your name, how you use AI, and one recent surprise.

INTRODUCTIONS

- 1 Name, organisation, one line on what you do
- 2 Share how you use AI: Style, Prompts, Topics, Tools, Products, Assistant, etc.
- 3 One recent surprise with AI, or one rule you personally follow

WHILE YOU LISTEN

Note the two people whose work sounds nearest yours. You will check that guess against your agent's answer in Exercise 3.

Note how people use AI different than you.

TAKE AWAY

How do people use AI similar to you? How do they use it differently?

AI is still early, like farmers adapting cars before anyone built a tractor.

1 · THE KIT

Iowa, 1915

A farmer jacks up a Model T's rear wheel to run a corn sheller. He wants the corn shelled, not the car.

2 · THE PATTERN

No correct way

Boulton & Watt sold engineering kits and instructions, not finished engines. Components meant to be tinkered with.

3 · THE PRODUCT ARRIVES

Then the tractor

Ford watched those uses and built the product. The kit stage never shows up in the statistics — and that is where AI is now.

This is not workflow optimisation, build-a-website, or prompt tricks. The tools that hide the assembly from you are not it either.

§ KIT PHASE

The stage of a technology when people are adapting loose parts to their own jobs and nobody has built the standard product yet.

Five maturity levels run from invisible personal use to civilization scale.

GOVERNED — WHAT THE ORGANIZATION APPROVES		CULTIVATED — WHAT ITS PEOPLE INVENT	
L·01	SHADOW AI in use through personal accounts; no monitoring and no governing policy.		PLAY Curiosity-driven exploration; AI met as a medium for deep play and discovery.
L·02	SANCTIONED Access policy in place; tools approved, use tracked, risk contained.		KIT — WE ARE HERE Users encoding personal workflows into reusable, shareable protocols of their own design.
L·03	ENGINE Workflows built around AI with named owners and quality metrics.		PRACTICE New forms of work emerge that were not possible before AI; roles are redefined.
L·04	INFRASTRUCTURE AI is a sector baseline; the coordination problem crosses organizations.		VERNACULAR Practices absorbed into culture; tacit knowledge spreads without mandates.
L·05	PLANETARY Civilization-scale AI; coordination and legibility become the governance challenge.		FLUENCY AI fully absorbed into how people think and work — second nature, no longer named.

A kit customizes a new general-purpose technology before it settles into products.

Loose parts you assemble and adapt to your own job, while the form is still open. In AI: the prompts, files and steps you bent around one recurring job.

The folder shape is an open standard, so a kit you build today still works in whatever assistant you use next month.

A **factory** — tonight's topic — runs without you, bends to its user, and hands off work others can build on.

YOUR KIT FOLDER

`README.md` — for people

`SKILL.md` — for the assistant: what the job is, when to use it

`examples/` — one worked input and output

`LOG.md` — what changed, and what broke

Template in the repo: `PARTICIPANT_TEMPLATE/`

§ KIT Customizing a new general-purpose technology before it settles into products — components meant to be tinkered with, with no single correct way to use them.

What happens after Kits?

DISCOVER → ENCODE

Kit

Custom practice

STANDARDIZE · THE FRONTIER

Factory

Scaled practice

NETWORK

Bridge

Connected practice

§ MATURITY MODEL

A ladder of levels describing how far an organization has got with something, used to locate where you are and name the next step.

What interfaces are emerging to access AI-native Kits and Factories?

THE CORPUS

The Protocol Institute archive of essays, videos, content, discussions, training, readings, and more...

DOOR 1 · PEOPLE

A web page — c3po.protocolized.io

DOOR 2 · COMMUNITY

A Discord bot — @C3P0 in the Institute server

DOOR 3 · MACHINES

Any assistant can call it directly (MCP):

```
claude mcp add c3po --transport http
https://c3po.protocolized.io/mcp
```

**MCP**

A standard that lets an AI assistant call an outside service directly, with no person copying anything across. Short for Model Context Protocol.

How can you, and managed AI agents, signal to others?

A trace left in a shared medium prompts the next action — no plan, nobody online at the same time.

SEMATECTONIC

The work itself

Your kit

MARKER-BASED

Signs about the work

Your LOG.md

THE MEDIUM

The production system

The project's infra

§ TRACES (STIGMERGY)

Coordination through what people leave behind rather than what they say: a trace in a shared place prompts the next person's action, with nobody present at the same time.

Stigmergic collaboration requires public signalling

WHERE AI PRACTICE SITS TODAY

Journal

One observer, kept to yourself. It works, and it dies with your laptop or the walled-garden platform



WHAT WE ARE DOING TODAY

Field notebook

One observer, exposed for others — Humboldt recording observations nobody asked for, in a form others could use.

Many observers, interior: a novel. Many observers, exterior: stigmergy, which we have just described.

Your inventory is your field notebook.

AI lets you run your thinking like a smallholding: automated, customised, and local.

Rao calls this cognitive farmsteading, and says it rests on three affordances of AI.

AFFORDANCE 1

Automation

You set up a whole intellectual farmstead — a dizzying variety of automations and scaled operations, run by very few people.

AFFORDANCE 2

Bespokification

You create $n=1$ products fitted to your own needs, like a farmer improvising equipment. Your thinking gets more idiosyncratic and better adapted.

AFFORDANCE 3

Pensievification

You share the raw outputs with a few trusted people — like trading excess produce with neighbours. It asks for high trust and it builds it.

Lean into all three and the tendency is **divergence, not convergence**. Your practice becomes better adapted to you — and stops making sense to anyone else.

A shopping trolley is fine in the shop and useless in someone else's kitchen. Rao's own early answer is the one we will use tomorrow: humans talking through what their agents produce and consume.

AI is a shared substrate ("the model"). This means everyone arrives at the same idea, often at similar times

THE EFFECT

Anything worth thinking about is being thought about by everyone capable of it, with AI in the loop — down to similar metaphors and names. There is no gradient left for a new idea to flow along.

Rao: a great many thoughts are now "common knowledge at birth".

IT REACHES QUESTIONS TOO

Models behave like index funds not only of past knowledge but of present curiosities. Explore an unformed question and others converge on the same framing.

So being first to an idea has stopped being the game.

KEY QUESTION: WHAT IS SCARCE INSTEAD?

§ PRECOMMODIFICATION

Sachin Benny's term: the model has already absorbed the archive, so having the idea is not the scarce part. What is scarce is the record of where it came from.

TBD

PENDING
PENDING

	ARCHIVE — WHAT IS ALREADY THERE	LIVE — WHAT IS HAPPENING NOW
EXTERNAL — THE ENVIRONMENT	Precommodification patterns Provenance is scarce, not novelty — the idea is already in the model	Traces / pheromones What other people's assistants are leaving in shared files, repos and channels
INTERNAL — YOU	Anchored in reality Your data, your constraints, your domain knowledge — the check against confident fiction	Playability You can act in the space, not only take things out of it

We come back to this grid at the end of every session: which quadrant did the last hour expand?

Ask your AI assistant to list the AI kits you already use at work.

STEPS

- 1 Open your harness at the top of the clone
- 2 Paste the prompt on the right and say "do this"
- 3 Answer its questions — it will not invent kits for you
- 4 Let it commit and push, then check the file on GitHub

Counts as a kit: a saved prompt, a custom GPT or project, a doc you copy from, a sheet with an AI step, an automation, a skill. One to three is plenty.

PASTE THIS TO YOUR ASSISTANT

Read `instructions/S1-inventory.md` in this repo and do this. Interview me one question at a time about the AI kits I already use at work. Do not invent any. Write the result to `participants/<my-handle>/s1-inventory/inventory.md`, then commit and push.

DONE WHEN

`participants/<you>/s1-inventory/inventory.md` is on main and you can open it in a browser.

§ AI ASSISTANT (HARNESS)

The program you talk to, which runs the model and reads, writes and runs things on your computer. Claude Code and Codex are the two this workshop uses.

Ask it to find the two people whose kits sit closest to yours.

STEPS

- 1 Pull, so you have everyone's inventories
- 2 Paste the prompt and say "do this"
- 3 Write the two names down — they seed tomorrow's groups

Read through your agent, not your browser. That is the habit the whole of tomorrow depends on.

PASTE THIS TO YOUR ASSISTANT

```
git pull, then read every participants/*/s1-  
inventory/inventory.md. Name the two whose kits  
are closest to mine and say in one line why for  
each. Append the answer to my inventory.md under  
"Nearest", then commit and push.
```

DONE WHEN

You have two names, with a reason each, pushed to your own folder.

§ INVENTORY Your own list of the AI kits you already use, one to three of them. It is the first thing you put in the repo.

Share one thing you noticed this hour, and together they show what we are learning.

Which of the four did this hour expand for you? There is no right answer, and nobody will be corrected.

EXTERNAL · ARCHIVE

Precommodification
patterns

EXTERNAL · LIVE

Traces

INTERNAL · ARCHIVE

Anchored in reality

INTERNAL · LIVE

Playability

Answers are tallied across all four sessions into the room's own awareness profile, read back in S4.

Stuck? Let's troubleshoot it together

ASync 1 — DELIBERATELY SMALL

- 1 Choose the kit you will turn into a factory tonight — yours, or one of ours
- 2 Let your assistant draft the README's Job and Interface lines
- 3 Push, and stop there

PASTE THIS TO YOUR ASSISTANT

Read `instructions/async-1.md` and do this. Ask me which kit from my inventory to convert, then draft only the Job and Interface lines of `participants/<my-handle>/s2-factory/README.md`. Commit and push. Stop there.

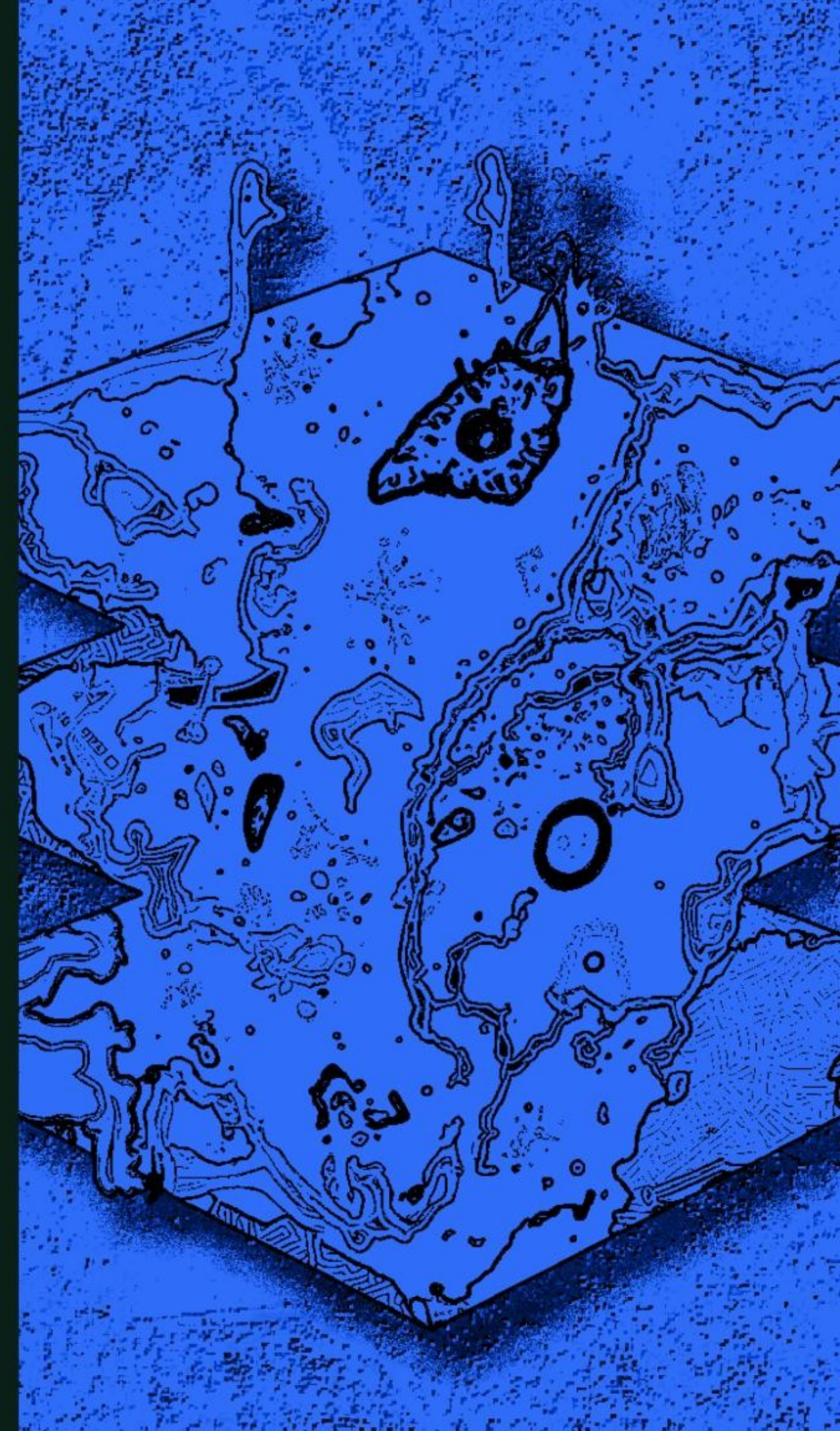
TONIGHT

Tech Support 16:30–17:00 UTC in `#kafka` — anything that will not log in, clone or push. S2 at 20:00 UTC, and that is when you build the kit.

SESSION 2 · 60 MINUTES

Factories

Tonight your kit learns to run without you. You leave with a folder another person's agent can run cold.



Kits running without their author's oversight need hard protocols to stay reliable.

RULES WRITTEN AS ADVICE

Rules that exist only as written advice. A kit works because its author re-bends it every single time.

RULES BUILT INTO THE PROCESS

Constraints promoted from advice into physics: a fixed output shape, a worked example, checks it runs on itself.

Tonight you remove yourself as something the kit requires. Letting a kit run more independently is only safe when its rules are built into the process rather than written as advice.

Housekeeping: recording on, consent as pinned, breakouts may be recorded · the S1 awareness check is read back exactly as it was counted · and the kits chosen in Async 1 are counted from the repo before we start

§ HARD WORLD / HARDNESS Jay Springett's sense: rules built into the surroundings themselves, rather than left as written advice an assistant may or may not follow. Hard does not mean difficult.

In the chat, write one thing your kit still needs you to do.

POST IN #WORKSHOP-KITKRAFT

"My kit still needs me to ___"

One line, no names, no explanation. Don't read the others until you have posted yours.

EXAMPLES, TO UNSTICK YOU

...pick which file to start from · ...know which client this is for · ...notice when the output is wrong · ...remember the exception we agreed in March

Why we start here: whatever you type is the thing your SKILL.md does not say out loud yet. Keep the line — you will use it in Exercise 6.

DONE WHEN

Your line is posted in the channel before you read anyone else's.

Electricity paid off only when factories rearranged around it, and AI is the same.

GROUP DRIVE · 1890S

One big motor on the old line shafts

Steam engine out, electric motor in, every belt and shaft exactly where it was. Almost nothing happened for years.

UNIT DRIVE · THE PAYOFF

A motor on every machine

Now the floor can be laid out by the work instead of by the shafts. The gains arrived with the rearrangement.

Most of us have done the first thing: put AI inside the workflow we already had. Tonight you re-lay one small floor — at the size of one person's practice.

§ WORKFLOW The ordered steps by which a piece of work actually gets done. Dropping AI into an unchanged workflow is the common move, and the one that changes least.

A factory runs without you, fits whoever uses it, and hands off useful work.

Four properties — and each one comes from somebody different. Nothing here says folder, app or website.

FROM THE AUTHOR

Reproduction

The steps, written down,
so it gives the same result
each time

FROM THE USER

Personalization

Their data, constraints and
context, supplied when
they run it

FROM THE FACTORY

Independence

It asks for what it needs,
instead of you explaining

FROM THE OUTPUT

Composability

Shaped so another factory
can use it unchanged

If any one of the four is missing you have a kit, a widget, a service, or a tool. Use the four on your own kit tonight.

§ FACTORY A kit that runs without its author, bends to its user, and hands off something others can build on. Defined by what it does, not what it is made of.

Two real examples show work that runs without its author and helps others build.

RAO → DIXON

The manuscript factory never made a book. It made drafts with metadata and left them in a shared folder; Jenna Dixon's production factory picked them up with no explaining.

Connects two factories by agreeing on an output — not by making them alike. Neither is a coder.

THE PI BRAND KIT

An assistant reads `brand.json` through `llms.txt` and has the brand, with nobody in the middle.

Monstrous Times copied it and bent it to its own world — that is forking, the property the book case does not show.

READ BOTH AGAINST FOUR QUESTIONS

Same result each time? · Bends to its user? · Runs without its author? · Hands off something to build on?

What held Rao and Dixon up was never machinery. It was agreeing how one practice meets another.

Write it down, break it into steps, name what it produces, and let others test.

1 · WRITE DOWN WHAT YOU DO

Write down what you actually do, including what you do without thinking

Taylor 1911 · Polanyi 1966

2 · BREAK IT INTO STEPS

Each step becomes a workstation — the numbered steps of SKILL.md

Smith 1776, the pin factory

3 · NAME WHAT IT PRODUCES

The intermediate that leaves is the thing someone else builds on

Chandler 1977

4 · PUBLISH WHAT OTHERS CAN RELY ON

Say in public what others may rely on: Exports: Interface: Adapts to: Hounshell 1984 · Baldwin & Clark 2000

5 · LET SOMEONE ELSE TEST IT

Someone else's assistant runs it without you in the room

Exercise 7, in 25 minutes

6 · KEEP THE ENTRY POINT STABLE

Keep the entry point unchanged while you change what is behind it

Parnas 1972

§ TACIT KNOWLEDGE

What you know how to do but have never put into words. It is why someone else's assistant stalls at the step that seemed too obvious to write down.

A factory can be a written guide, a script, an app, or a website.

1 A skill — a README and SKILL.md an assistant runs, plus one example
Standard: SKILL.md / Agent Skills · tonight, everyone is here

2 A single file that calls a model through MCP or an API
Standard: MCP · modelcontextprotocol.io

3 An app someone installs
More reach, more effort — not more of a factory

4 A website running on a machine somewhere
Standard: A2A, agent to agent

The definition is the same on every rung. Standards that close early lock in whatever happened to be there — which is how we all ended up typing on QWERTY. Publish the smallest possible entry point, and keep everything behind it free to change.

§ STANDARD A protocol written down and published so anyone can build against it. The ones named here get named, not taught.

Turn your kit into a factory someone else's AI assistant can run without asking you.

PASTE THIS TO YOUR ASSISTANT

Read `instructions/S2-factory.md` and do this. Walk me through the six steps in order for the kit I chose in Async 1. Ask me for anything you need – do not invent answers. Write everything to `participants/<my-handle>/s2-factory/` and push.

FOUR THINGS MUST EXIST WHEN I CALL PUSH

`README.md` — three lines: Exports, Interface, Adapts to

`SKILL.md` — numbered workstations that ask for the user's context, not yours

`examples/` — one output your assistant produced cold from `SKILL.md` alone

All of it inside `participants/<you>/s2-factory/`

DONE WHEN

A stranger's agent could open the folder and produce the example without messaging you.

A PRACTICAL TIP

Fix the output *shape*, not just the intent: column order, units copied never converted, a required reference.

§ SKILL A folder holding a `SKILL.md` file that tells an assistant how to do one job, plus anything it needs. Its shape is a published standard, so it works in another assistant.

Swap with a partner and see where their assistant gets stuck or gets it wrong.

TEST 1 · RUNS WITHOUT ITS AUTHOR

Your assistant gets the README and SKILL.md and nothing else. Can it produce the example without asking the owner a single question?

TEST 2 · FITS ITS USER

Give it one line about your own situation through Adapts to: — does the output suit you, or still suit the owner?

PASTE THIS TO YOUR ASSISTANT

Read only participants/<THEIR-HANDLE>/s2-factory/README.md and SKILL.md. Run the example using nothing else. Then run it again with this about me: "<one line of my context>". Report in three lines per test what you could not do.

HOW TO RUN IT

Read in a ring — the pin says who reads whom

Read your three lines aloud to the owner

The owner types nothing back except blocker lines in their LOG.md

DONE WHEN

Your partner has three lines per test, and every failure is a blocker line in their LOG.md.

Share one thing your partner's assistant could not do without you.

FINISH THIS SENTENCE

"The other agent got ___ wrong about my kit"

Most answers will be one of these two. Both are the next line of your kit:

UNWRITTEN KNOWLEDGE

It stalled where you know something you never wrote down

FITTED THE WRONG PERSON

It ran fine and produced something that suited you, not the person using it

DONE WHEN

Every person has named one thing, and each owner has it written down as their next line.

Share one thing you noticed this hour, and together they show what we are learning.

Same question as this afternoon — which of the four did this hour expand?

EXTERNAL · ARCHIVE
Precommodification

EXTERNAL · LIVE
Traces

INTERNAL · ARCHIVE
Anchored in reality

INTERNAL · LIVE
Playability

Overnight, fix what your partner found and save the new version.

WHAT TO DO

- 1 One fix per blocker line — could not run without you, or did not fit the reader
- 2 Re-run the example cold
- 3 Push v2 with one progress line saying what changed

PASTE THIS TO YOUR ASSISTANT

Read `instructions/async-2.md` and do this. For each blocker line in my `LOG.md` make exactly one fix, re-run the example cold, then push v2 with a one-line progress note.

DOING TONIGHT ASYNC?

Push your factory anyway — your assigned reader's agent will test it and post the report in the channel for you.

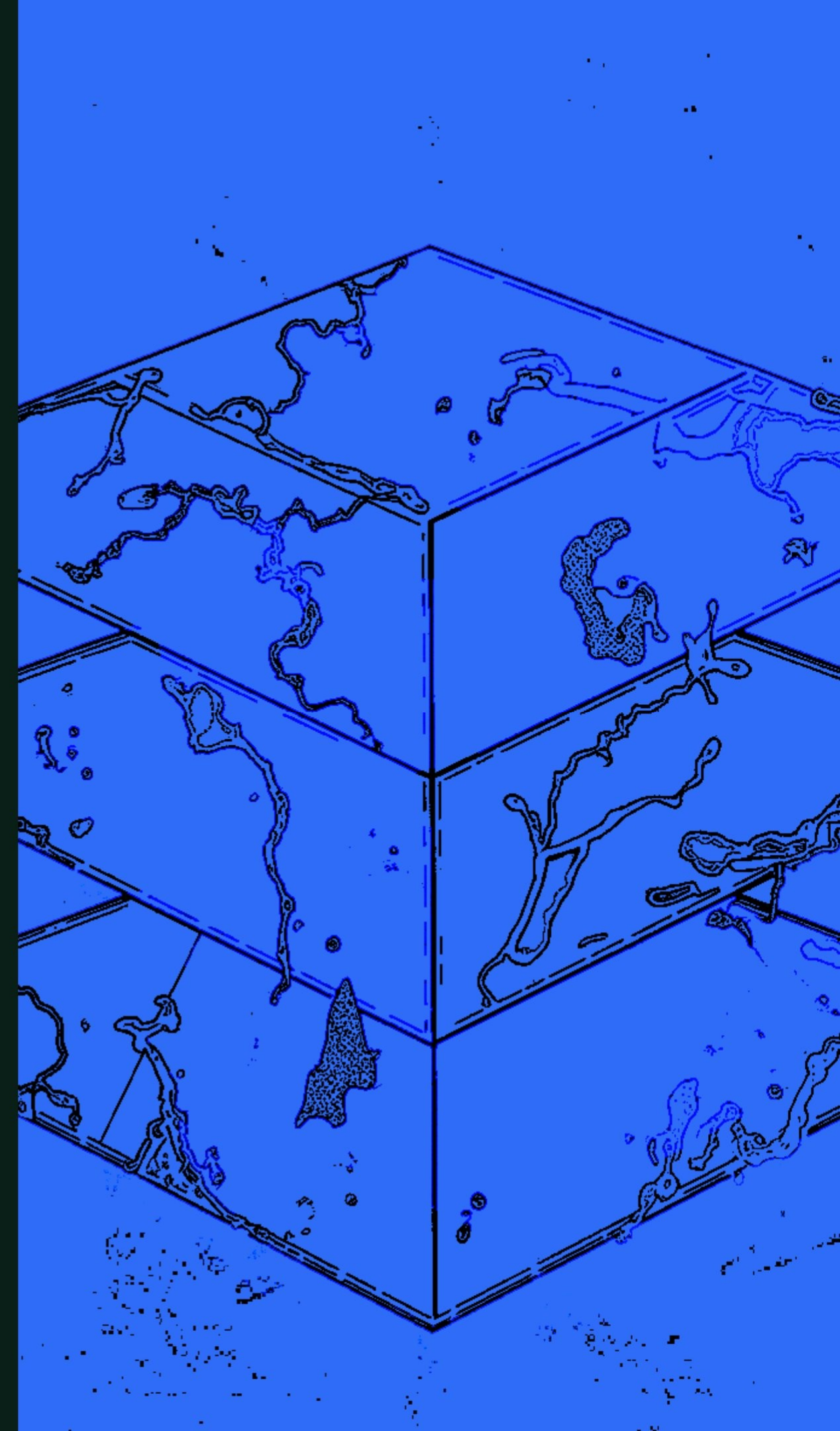
S3 tomorrow 15:30 UTC — bridges. You will use someone's output unchanged instead of copying it. Bring the kit you did *not* critique in mind.

§ PUSH Sending your commits up to GitHub, where everyone else can see them. Until you push, your work exists only on your machine.

SESSION 3 · 60 MINUTES

Bridges

One AI assistant can use several kits together, reliably.
How do two private practices reach each other without a
shared app?



Yesterday's factories each work alone, but none of them has been combined yet.

IN THE REPO NOW



EVERY KIT HAS A FAILURE LOGGED

And almost all of them are description problems, not code problems.

WHICH MAKES THEM USEFUL TODAY

A failure is a trace: it tells the next agent exactly where the knowledge was still in somebody's head.

Kits that landed async overnight count — their bridger is the first cold reader.



The running list of one-line notes your assistant writes in LOG.md as it works. It only ever grows: earlier lines are never rewritten or deleted.

In the chat, name two kits in the room you would like to combine.

POST IN #WORKSHOP-KITKRAFT

"___ plus ___, so that ___"

Two folder names and one reason. Browse participants/ if you need a reminder of what exists.

USEFUL PAIRS ARE USUALLY UNEVEN

One kit that *produces* something plus one that *needs* that thing — not two kits that do the same job.

Nothing is assigned on the basis of this. Your target is in the pinned post — but a pairing named here can be swapped in.

DONE WHEN

Your pairing is posted, naming two folders and one reason.

A bridge is a kit between kits, running several of them as one process.

A target is any kit or factory in the repo. The difference is one line: a factory declared an Interface: in S2, a kit did not.

FACTORY → FACTORY

Your agent uses the target strictly through its declared interface

KIT → KIT

The interface is the README a person hands to their agent

KIT ↔ FACTORY

One side declared; the log records whether that way in was enough

Three configurations to try, **not a taxonomy** — your harness works out which one you are in and writes it down.

§ BRIDGE One AI assistant using several kits or factories together and making that a reliable process; a kit between kits. What crosses is usually half-finished work.

Passing half-finished work between kits often works better than passing finished results.

THE PROBLEM, RESTATED

Bespoke use makes tooling private; precommodification means we all reach the same ideas; farmsteading keeps each practice closed.

SO SHARING RUNS ON

Public records and interfaces — not on everybody adopting the same app.

RAO'S "INVISIBLE 90%"

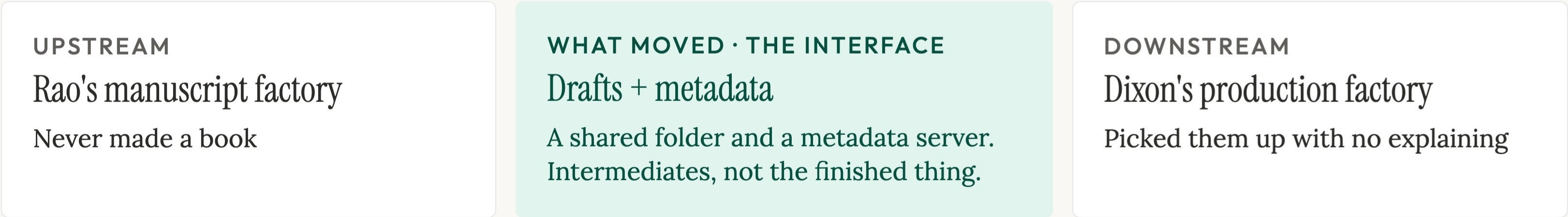
High-trust links passing intermediates between bespoke setups. Skills are industrial intermediates — **the kit you wrote last night is one.**

Unix made this its whole design: expect the output of every program to become the input to another, as yet unknown, program.

§ INTERMEDIATE

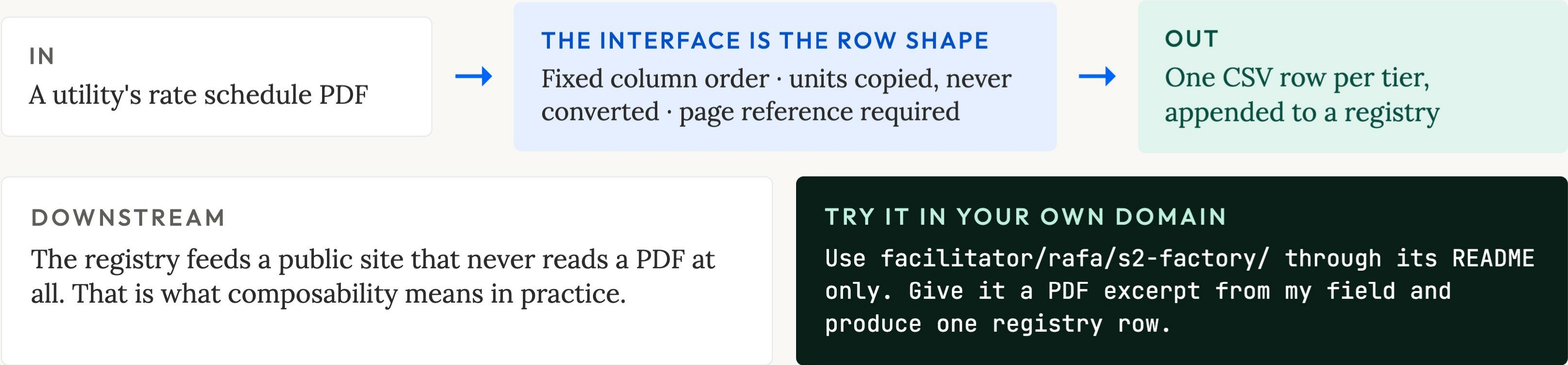
The half-finished thing one kit hands to the next: a draft, a set of rows, a file with notes attached, rather than a finished product.

Rao and Dixon linked their setups by passing drafts, not finished books.



Two non-coders. Domain knowledge beat coding knowledge, because the hard part was never machinery — it was agreeing what crosses the gap.

One assistant can chain two kits, feeding the first kit's draft into the second.



§ INTERFACE The way in: what someone else needs to know to use your kit without asking you anything. It can be as plain as the column order of a row.

A combined process is only useful if it runs the same way next week.

The paragraph you write at the end of the breakout is the data. Tomorrow we read all of them before anyone says anything. While you work, watch for four things:

1
What did your agent need from the target?

2
What did it find on its own?

3
What did a human have to supply?

4
What stayed fixed between the two sides?

Everything on the last few slides is evidence to check your own log against — not an answer to match.

Using someone's instructions means trusting them, so check them before you rely on them.

WHY THIS HAPPENS

A loaded `SKILL.md` arrives as instructions, not documentation — and the agent cannot reliably tell a third party's instructions from yours.

WHAT THE RESEARCH NOW SAYS

Skills are dependencies; treat them as untrusted by default. That is the formal statement of why bridges are high-trust.

26.1%

of 31,132 public skills analysed carried at least one vulnerability.

Agent Skills in the Wild, 2026 · CSA research note, May 2026

This room runs on trust, so no read-before-load rule is imposed today. That is a property of this room, not of the world — which is why identity and trust standards are being built at all.

Ask your assistant to combine two or more kits into one repeatable process.

PASTE THIS TO YOUR ASSISTANT

Read `instructions/S3-bridge.md` and do this.
TARGET: `participants/<their-handle>/s2-factory/`
MY KIT CANNOT: `<the one thing>`
Use the target only through its declared interface.
Log every human unblock as one line.

HOW TO WORK DURING THIS EXERCISE

agent → repo → agent

Talk to a human *only* to unblock, and every unblock becomes one line in your log. The block refuses to invent the task — that part is yours.

YOUR TARGET

The kit you did *not* critique — or swap to a facilitator kit

AT MINUTE 30

Stop building. Write the two paragraphs, even if it half-worked

DONE WHEN

A bridge log is pushed, ending with "what a bridge turned out to be"

§ GRAPH

Several steps laid out in a fixed order, saying which runs next and what it hands on. A bridge is a small graph; a loop walks each step of it.

Share what broke when kits were combined and what made the process repeatable.

FILL IN THE SENTENCE

"My agent needed ___ from the target and found ___ on its own"

The long version is already in your log — this is the headline only.

Order is posted in [#workshop-kitkraft](#). A half-worked bridge is a real answer.

DONE WHEN

Every person has given one line, and the sentence is filled in.

Share one thing you noticed this hour, and together they show what we are learning.

Which quadrant did this hour expand?

EXTERNAL · ARCHIVE
Precommodification

EXTERNAL · LIVE
Traces

INTERNAL · ARCHIVE
Anchored in reality

INTERNAL · LIVE
Playability

Tomorrow's open question: what actually makes a combined process reliable?

HYPOTHESIS · SAID ONCE, TO BE TESTED AGAINST YOUR LOGS

A bridge is a protocol between two factories — the smallest thing both sides agree not to change.

Tomorrow we read your paragraphs first and this second, in that order. If they disagree, your paragraphs win.

§ **PROTOCOL** An agreed, reusable way of coordinating that people or programs can rely on without renegotiating it each time. It can be technical, like a file format, or social.

Before tonight, save your bridge notes and prepare your short show-and-tell.

TWO THINGS, IN ORDER

- 1 Finish the bridge log — including the "what a bridge turned out to be" paragraph
- 2 Have your harness prepare your ninety seconds for S4

The hard-mode rerun is optional and goes in the same file.

PASTE THIS TO YOUR ASSISTANT

Read `instructions/S4-show-and-tell.md` and do this. From my bridge log, draft `show-and-tell.md`: target, the one thing, what a human supplied, whether it worked. Ninety seconds spoken. Push it.

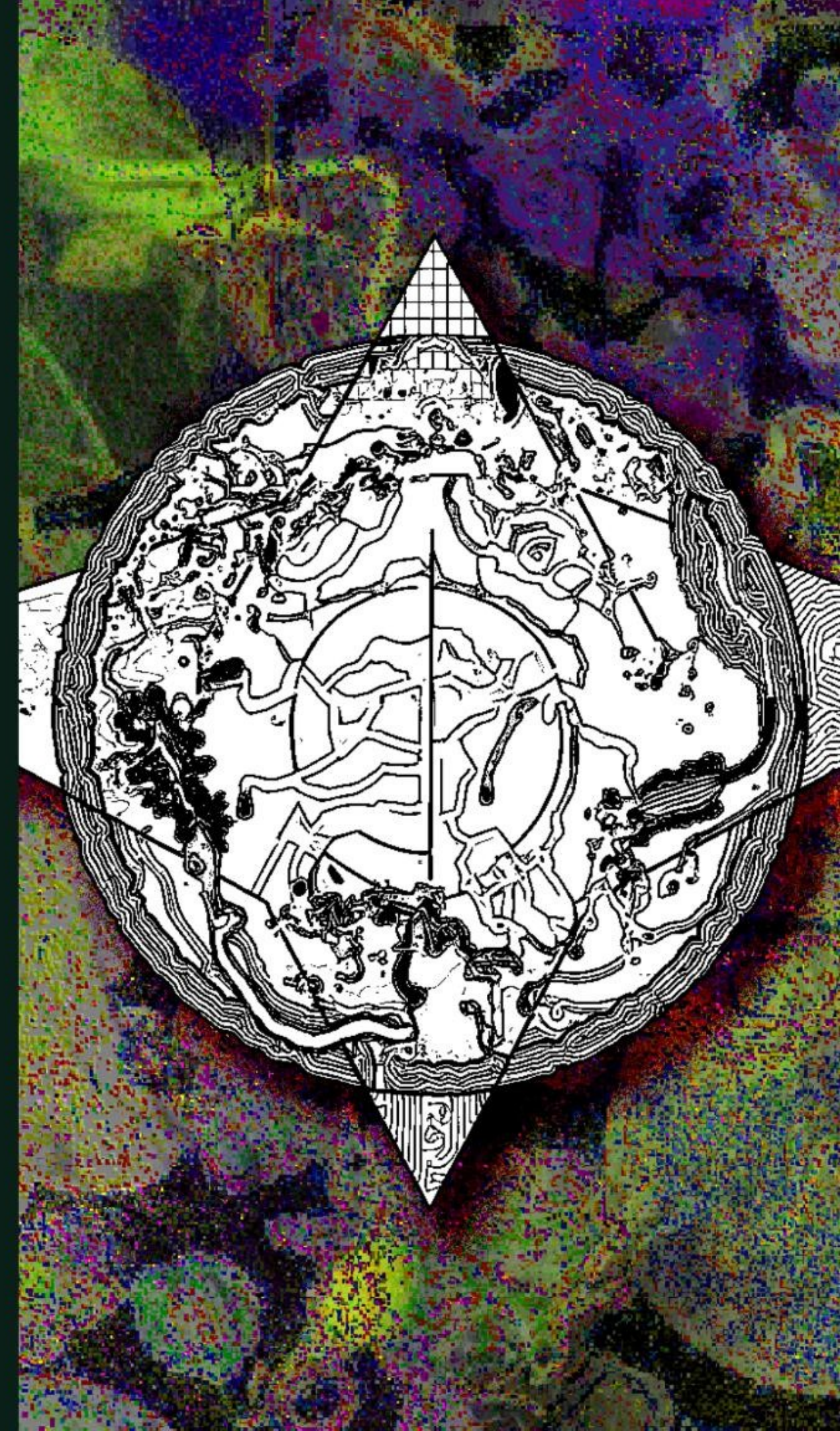
CAN'T MAKE S4?

Push the file anyway — a group-mate reads it aloud in sixty seconds, and your awareness answer is read from it too.

SESSION 4 • 60 MINUTES

Show and tell

Tonight the slides are the work you made. There are no slides of ours to show you.



Two days ago these were ideas, and now they are files anyone can open.

—
.....
inventories

—
.....
kits turned factory

—
.....
bridge logs

3
public transcripts

Two days ago every one of these was a task inside somebody's head.

Tonight there are no slides of mine to show you, because your files are the slides.

Recording on · consent as pinned · nothing tonight is off the record

§ REPOSITORY (REPO)

A folder of files whose whole history is tracked, with a shared copy on GitHub. Ours is public, and everything made in the workshop lands there.

In the chat, describe how your bridge went in one word.

ONE WORD IN #WORKSHOP-KITKRAFT

No sentence, no caveat, no explanation. The nine hundred words are already in your log — this is a quick sense of how it went.

DIRECT ANSWERS ARE WELCOME

"Broke", "surprising", "tedious", "nearly" are all useful answers. The bridge that failed is the most useful file in the repo tonight.

DONE WHEN

One word from every person is in the channel.

Three real files show the journey from a kit to a factory to a bridge.

KIT

SKILL.md v1 → v2

Monday's file beside the same file after the critique. The change is in the **description line** — where nearly all of them were.



Screenshot: SKILL.md v1 beside v2, description line changed
or [browse files](#)

FACTORY

README.md · Interface:

One line was the whole interface. Every bridge ran on **rung one** of the ladder — and that turned out to be enough.



Screenshot: a README with its Interface: line
or [browse files](#)

BRIDGE

LOG.md · one **unlock line**

Read aloud as written, because it names the thing that **was not in the repo**.



Screenshot: the LOG.md unlock line
or [browse files](#)

Each person gets ninety seconds to share their show-and-tell.

SAY FOUR THINGS, IN THIS ORDER

- 1 Your target
- 2 The one thing you made your agent do
- 3 What a human had to supply
- 4 Did it work

ON SCREEN

Your file goes up as your name comes round — you talk about your own file, not about slides

IF SOMEONE IS AWAY

A group-mate reads their show-and-tell.md in 60 seconds

TIMING

An agent captures every report into the synthesis as you speak

DONE WHEN

Every person in every group has been heard — present or read aloud by a group-mate — and each report is in the synthesis.

§ SHOW AND TELL The closing round in Session 4, where each person gets about ninety seconds on their own work with their own file on screen.

Two bridges that worked and one that did not teach us the most.

CHOSEN AT 19:45

The cleanest file handoff

CHOSEN AT 19:45

A kit → kit bridge

CHOSEN AT 19:45

The one with the most unblock lines

The one that did **not** work is the most useful file in the repo tonight — it is a precise list of what the next version of that kit has to contain.

Listen for the split: what the agent worked out on its own, and what a person still had to say out loud.

Your own words on what made your bridge reliable come first, read out verbatim.

FIRST The numbers, straight from the logs rather than from memory: bridges by kind, attempted and worked.

THEN The "what a bridge turned out to be" paragraphs, read as written, without paraphrasing any of them.

**DURING
THE
SESSION** If a paragraph is read wrong or a count is off, say so — it goes in under Corrections with your handle.

We agree as a room on what makes a bridge reliable, using your notes.

STATED ONCE, TUESDAY 16:28

A bridge is a protocol between two factories — the smallest thing both sides agree not to change.

SUPPORTS

Names a specific thing both sides held fixed

CONTRADICTS

Says a bridge was something else entirely

SILENT

Does neither

Every unblock line is something not yet agreed. What stayed fixed is the candidate smallest thing.

Verdict: retained, amended or rejected. Two thirds supporting and nothing contradicting to retain — ties go to the room.

Name what you now notice about your own work that you missed on Monday.

THE QUESTION THIS WAY OF PAYING ATTENTION DEPENDS ON
Monday morning, before any of this — which quadrant was your practice blind to?

- Precommodification
- Traces
- Anchored in reality
- Playability

Four checks in a row plus tonight is the room's own awareness profile. We record what it actually says — **including where it splits**. Absent answers are read from their show-and-tell files.

WHAT HAPPENS NEXT

The work stays public, and a summary arrives here within 48 hours.

THE REPOSITORY

The repo stays public
and yours

WITHIN 48 HOURS

SYNTHESIS.md v2, with
tonight's transcript

OPTIONAL

Hard mode: rerun your
bridge with no human
chat at all

CONTINUING AFTERWARDS

SIGBIZ fortnightly show-
and-tell — bring the kit,
not the slides

Recordings may feed Symposium write-ups and later research — tell either of us if you would rather they did not.