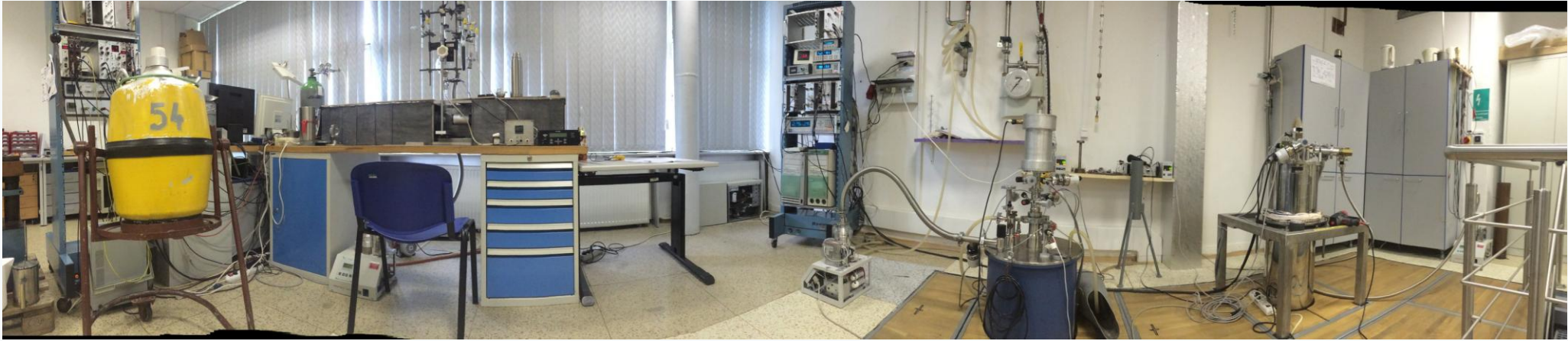


How the Mössbauer lab runs today



Seven spectrometers CH1–CH7 measure non-stop, 24/7

DATA

- Run log in a shared sheet
- Paper logbooks 2010–2018
- Digital logbooks since 2019, structure shifting by era
- Spectra, calibrations scattered on lab PCs

INSTRUMENTS

- One vendor program per device
- Helium and nitrogen refills planned from experience
- History only in loose log files

ANALYSIS

- Spectra folded and fitted by hand in multiple SW
- Calibrations re-processed manually
- No complete, user-accessible data library

ADMINISTRATION

- Research service for three institutions: different forms, approvals and budgets
- Kept in Word and a personal spreadsheet

From a hand-run lab to a lab that runs, learns and documents itself

LABORATORY MANAGEMENT AND OPERATIONS

One source of truth
per project and per
measurement.
Phases and states
visible to everyone.

AUTOMATION

Collectors,
generators and
pipelines take over
the routine; users
keep the decisions.

LONG-TERM MONITORING

Years of cryostat
and calibration
history become
models,
predictions and
insights.

SELF-IMPROVING ALGORITHMS

Every prediction is
compared with
reality and the
model corrects
itself.

LESS ADMINISTRATION

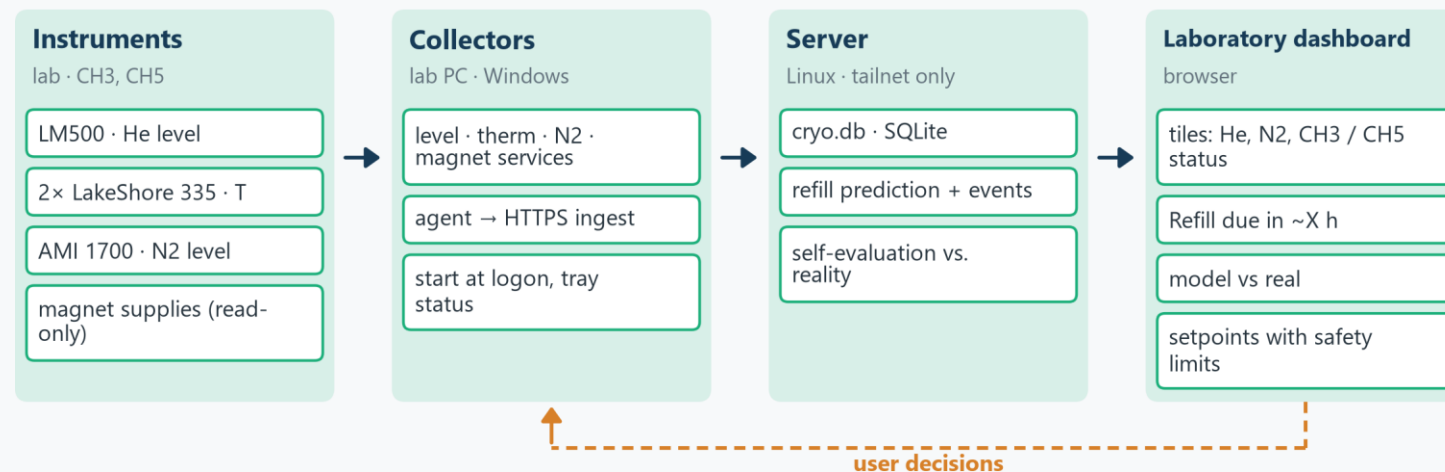
Forms, budgets
and reports
generated from
data entered once.

AI builds and runs the tools. The human stays the orchestrator.



- KFNT MFF UK

Cryostats: live data, prediction, remote control



WHAT IT IS

Live telemetry of the cryostats: helium level, sample temperatures, heater power and nitrogen, collected on the lab PC, stored on a server and shown in the Cryostat monitoring dashboard with a prediction of when helium runs out. The model is learned from the history since 2021 and checked against every new reading.

WHERE THE TIME IS SAVED

Nobody has to go to the lab to collect data points for a decision, or guess from experience. Lab know-how becomes data-backed, validated and transferable.

THE FIRST PROMPT IN FOUR LINES

GOAL Predict the helium refill and warn hours before a critical level.

CONTEXT Magnet and refill rules, event types in the data.

SOURCES He level log, LakeShore Chart Recorder exports for CH3 and CH5.

OUTPUT Dashboard: level and temperature tiles, refill prediction, model vs real.

THE SAME PROMPT, REWRITTEN AS A TASK FOR CLAUDE CODE

GOAL Predict when the CH3 helium cryostat needs a refill and warn the lab hours before a critical level. The system must be robust and never fail silently.

CONTEXT 0 cm = magnet centre at the sample. Magnet energised → refill at 5 cm latest. Full fill = 64 cm.

- Data gap > 72 h = shutdown, shorter = outage. Warm-up: He boils off, T rises toward 300 K.
- He level is read once an hour. Which LakeShore input holds the sample is declared, never guessed.

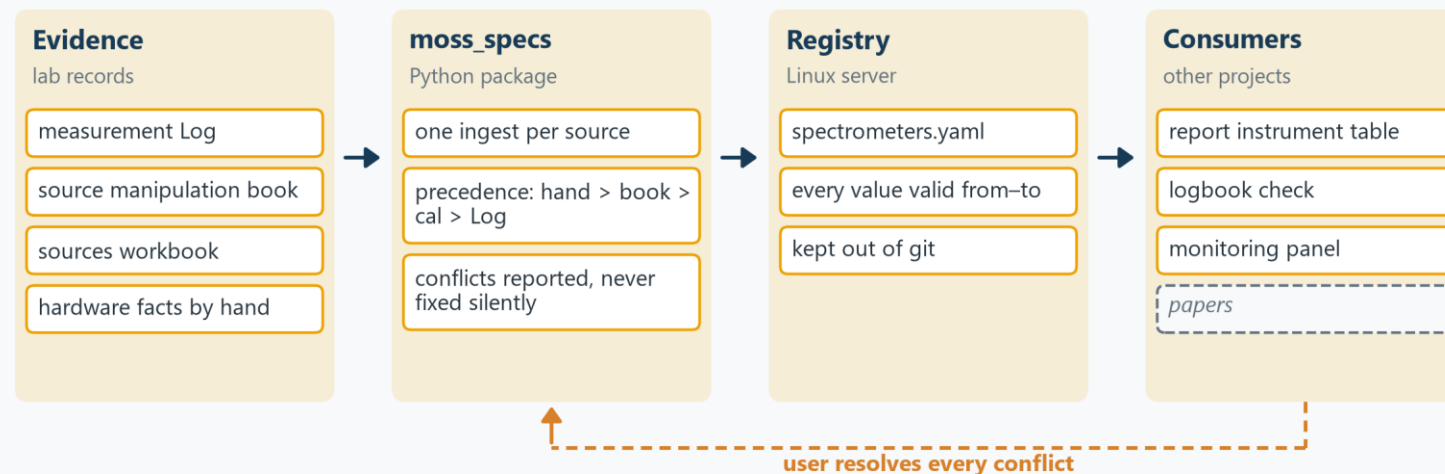
SOURCES He level log from level.py in MS_LHe_level/. LakeShore exports in MS_thermometers/.

- Export format: row 1 serial number, row 2 log start, column 1 = ms since start.

OUTPUT Dashboard Cryostat monitoring. Tiles: He level (+ estimate now), N2 level, CH3 / CH5 status (T, heater, field).

- CH3 graph: T and He level with events, model dots between readings. He tile: Refill due in ~X h.
- Model vs real for every reading. Missing data → NO DATA in the tile, never an old value.
- First read the data, list event types I did not describe, ask. No code before I approve.

Spectrometer specifications over time



WHAT IT IS

A registry of spectrometer specs and settings over time. Every attribute has its own validity, so one can ask "what did CH3 look like on a given day" - exactly what a report about a past measurement must describe. Hand-entered facts beat derived ones, and conflicts are reported, not hidden.

WHERE THE TIME IS SAVED

The instrument table of a report is no longer reconstructed from logbooks and the Log for every project. One query returns it, with conflicts flagged.

THE FIRST PROMPT IN FOUR LINES

GOAL Spectrometer specs valid from-to, read automatically into reports.

CONTEXT A spectrometer is a set of facts in time; the book beats the Log.

SOURCES Log, calibrations, manipulation book, sources workbook.

OUTPUT Registry + specs_at(channel, date), conflicts listed for the user.

THE SAME PROMPT, REWRITTEN AS A TASK FOR CLAUDE CODE

GOAL Build a registry of spectrometer specs valid from-to, so instrument tables in reports and papers fill themselves.

CONTEXT A spectrometer is a set of facts valid in time, not one record. Source, reverse and calibration change.

- Source moves: the manipulation book is primary, the Log secondary. A source sits on one channel at a time.
- Calibration CnAxxx belongs to channel CHn.

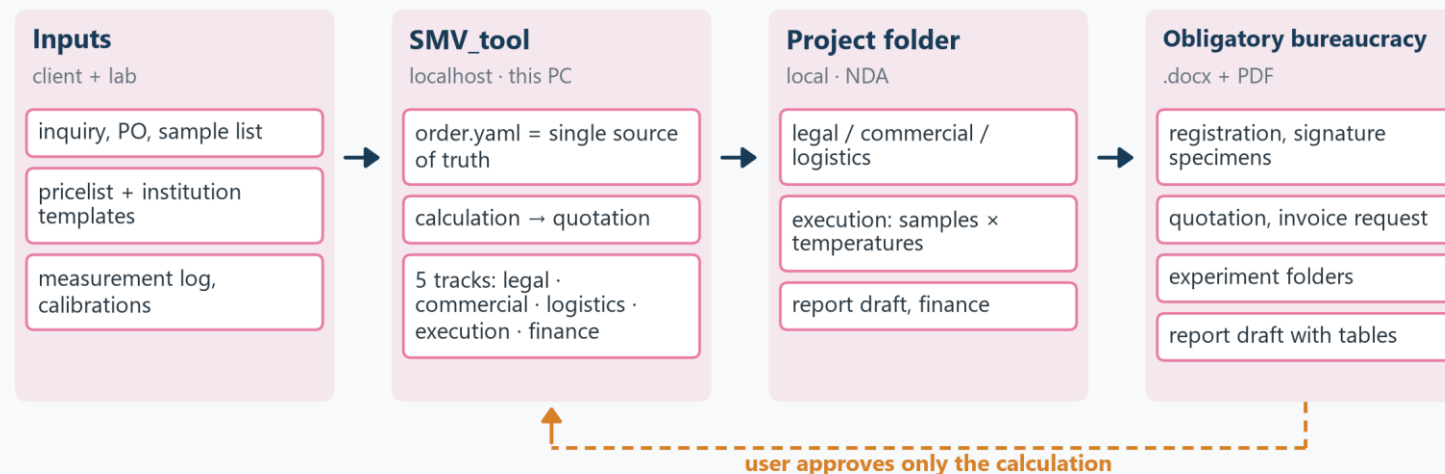
SOURCES cal_new.xlsx and the Log in MS_calibrations/.

- Photos of the manipulation book (OCR locally only). Sources workbook (path in .env).

OUTPUT Package moss_specs/: one ingest per source, precedence hand > book > calibration > Log, specs_at(channel, date).

- Report conflicts and gaps. Never guess, never overwrite.
- Source data never leave the lab: registry gitignored, show.py --mask for anything printed.
- Done when the book-vs-calibration check lists every disagreement for me to decide.

One project, one place, every document generated



WHAT IT IS

Automates the life cycle of a contract-research project - inquiry, calculation, quotation, registration, handover protocol, invoicing - for two institutions. A project is never in one state: five tracks run in parallel, so the tool is a consistency radar, not a gatekeeper.

WHERE THE TIME IS SAVED

Project data are entered once. Forms for each institution, the folder structure and the report draft are generated instead of retyped from the previous project.

THE FIRST PROMPT IN FOUR LINES

GOAL Carry a project from inquiry to invoice with all papers per phase.

CONTEXT Two institutions, different forms; steps overlap and may swap.

SOURCES Training folder of real filled forms per institution.

OUTPUT order.yaml, dashboard with phases, document generators, sample status.

THE SAME PROMPT, REWRITTEN AS A TASK FOR CLAUDE CODE

GOAL Take a contract-research project from inquiry to invoice and generate every document each phase needs.

CONTEXT Shared lab of two institutions: same measurement, different forms, approvals and report headers. Start with one.

· Steps: inquiry → (NDA) → calculation + quotation → PO + registration → budget source → samples → report → invoice. Steps overlap and may swap order.

· Own project number with the year (26-NNN), linked to every ID that appears later.

SOURCES SMV_training/: real filled forms per institution. Calibrations from MS_calibrations/.

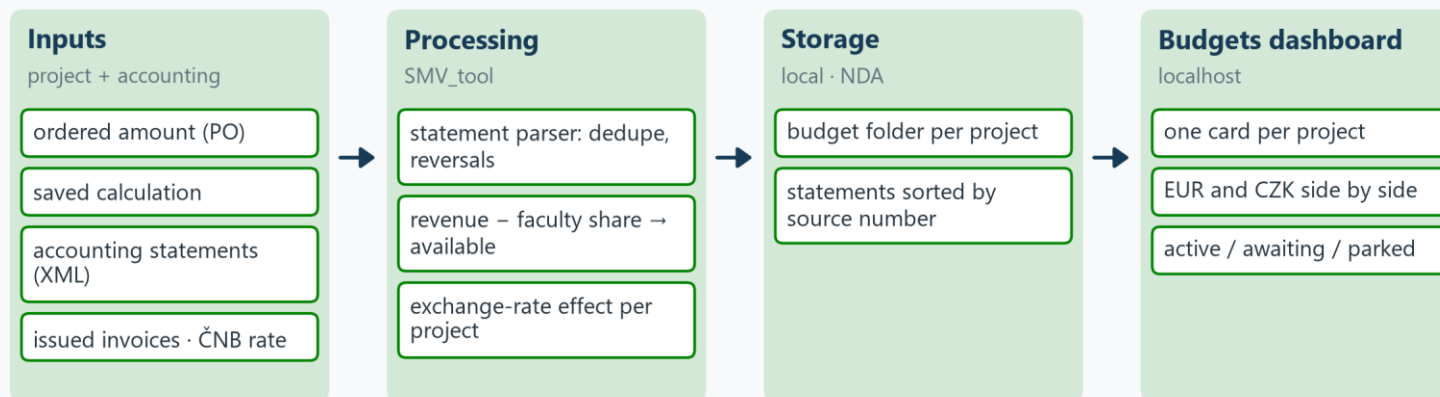
OUTPUT SMV_tool/: order.yaml as single source of truth, localhost dashboard by phase, generators → .docx + PDF.

· Sample status: measured / analysed / completed / problem.

· Client data stay on this PC: pre-commit check, nothing in git.

· First draw the process as a diagram for me to check, then ask what is unclear.

Budgets linked to accounting statements, every document generated



WHAT IT IS

The budget side of the same tool. Revenue, faculty deductions and spending per project are computed from the purchase order, the calculation and the accounting statements, at the right exchange rate for each moment.

WHERE THE TIME IS SAVED

Accounting statements are parsed instead of copied line by line into a personal spreadsheet. Reversals and exchange-rate effects are computed, not recalculated by hand.

THE FIRST PROMPT IN FOUR LINES

GOAL Budget and spending per project; rebuild this year's real budgets.

CONTEXT Rules per institution, exchange rate until invoicing, cryogens first.

SOURCES Projects, calculations, accounting statements, invoices.

OUTPUT Budgets dashboard, one card per project, accounting view first.

THE SAME PROMPT, REWRITTEN AS A TASK FOR CLAUDE CODE

GOAL Show budget and spending for every project and rebuild real budgets for this year's projects.

CONTEXT Spending rules differ by institution. The faculty takes overhead and a fixed share that must stay untouched.

- CZK revenue from EUR invoices: ČNB rate of the previous day until invoicing, then the rate on the invoice.
- Cryogens are paid first. Planned salaries are not part of the budget.

SOURCES order.yaml and the saved calculation of each project.

- Accounting statements (XML) per budget source. Issued invoices (PDF).

OUTPUT Budgets dashboard in SMV_tool: one card per project, EUR and CZK side by side (never summed), cryogens on their own line.

- Accounting view first, cost view later.
- The balance must match the accounting summary. Show mismatches, never hide them.
- Local only (NDA). Ask me about every rule you are not sure of.

A writing assistant grounded in my own papers and data

Example output: figure for a manuscript in preparation

THE FIRST PROMPT IN FOUR LINES

GOAL Help write papers and make publication figures from measured data.

CONTEXT 57Fe Mössbauer, fixed terminology, my own style; Word and Overleaf.

SOURCES Own papers, fit outputs, Scite.

OUTPUT Section corpus, CLAUDE.md, Scite search, figure pipeline.

THE SAME PROMPT, REWRITTEN AS A TASK FOR CLAUDE CODE

GOAL An assistant for writing Mössbauer papers (Abstract, Experimental, Results, Discussion) and publication figures from measured data.

CONTEXT 57Fe Mössbauer spectroscopy. Terminology δ , ΔE_Q , B_{hf} , A. Style learned from my own papers.

- I write in Word or Overleaf and need output for both.

SOURCES My older papers (.docx / .pdf) in _corpus/_inbox/.

- Fit outputs (SpectRelax, ConfitA) in _papers/<slug>/data/.
- Scite REST API, key in .env.

OUTPUT Corpus split into markdown sections + STYLE_NOTES.md. CLAUDE.md with role and rules.

- scite_search.py: DOI or keyword → supporting / contrasting citations.
- Figures: YAML recipe → I confirm → matplotlib PDF / PNG / SVG with provenance.
- Never invent a number: values only from data files. Manuscripts stay on this PC. Spec first, no code yet.

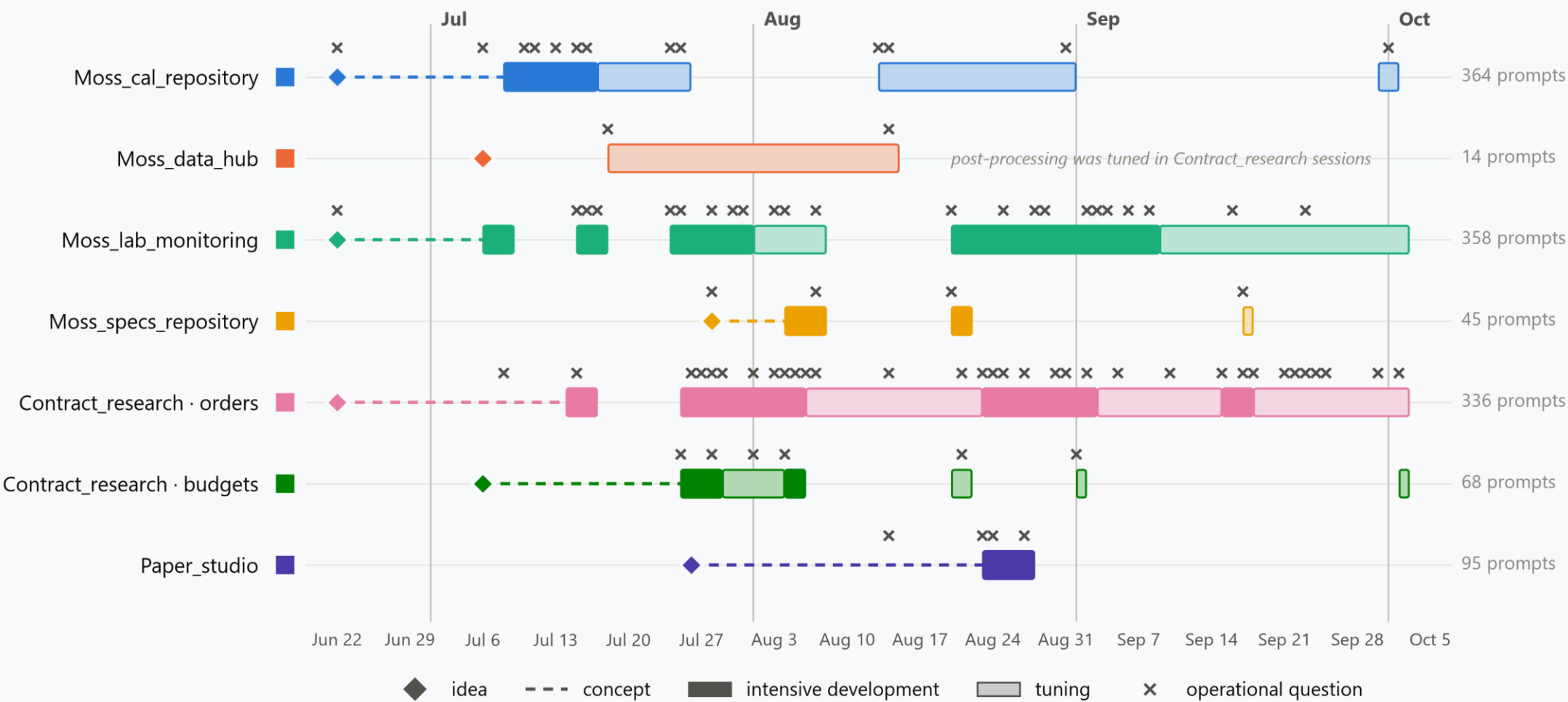
WHAT IT IS

An assistant for writing scientific papers and publication figures from measured data, driven by plain language in the terminal. One source is meant to yield LaTeX for Overleaf and .docx for Word. Runs only on this PC; data and manuscripts never leave it.

WHERE THE TIME IS SAVED

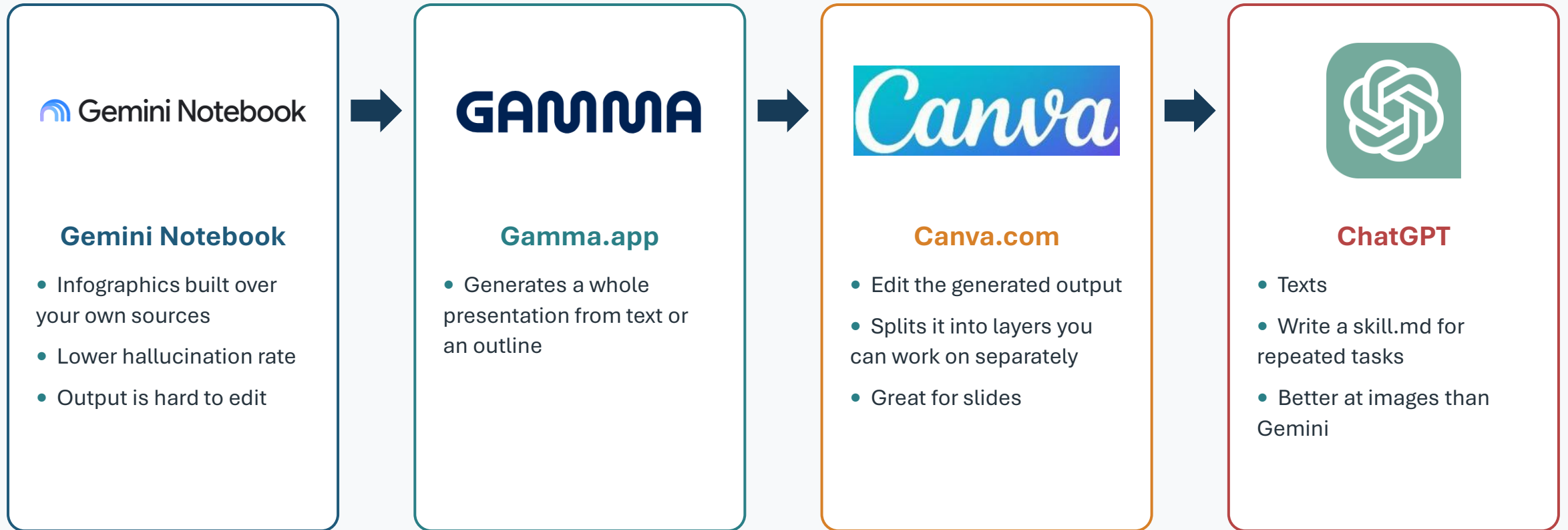
Figures are re-rendered from the data by script after every refit instead of being redrawn by hand. My own papers become a searchable reference for style and phrasing.

Four months of prompts: build, tune, use

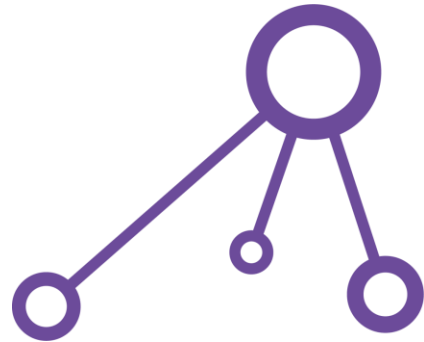


Every prompt typed into Claude Code since June, by project. Solid bar = intensive development, pale bar = tuning, x = an operational question to a tool that already works.

Useful non-CLI tools



Workshops and support



Leadership Academy for Researchers

Leadership Academy for Researchers

- Run by STAR Cluster with HiLASE Centre, Czexpats in Science and partners
- Pilot January–June 2026: six modules of webinars and hands-on workshops
- Leading people, teams and institutions in research

leadershipinscience.academy



Charles University • AI portal

- Generative AI tools for the university: M365 Copilot, AI sandbox, OneTutor
- Courses and seminars, e.g. M365 Copilot Chat for beginners
- Guides: data protection, safe use, AI disclosure for theses

ai.cuni.cz

Expanding support for scientists

- **Claude team plan for scientists:** 10,000 seats for one year. Standard seats free, premium seats with 5× usage limits for \$15 a month.
- **Who qualifies:** a principal investigator or equivalent at an academic or nonprofit research institution, through a verification form.
- **AI for Science:** up to \$50,000 in credits per project; the programme grows in scope and scale.
- **Claude Science:** a product that integrates researchers' usual tools, produces auditable artifacts and gives flexible access to computing.

Text, diagrams, charts and layout of this deck were generated with AI (Claude Code) from the author's materials.

Cost (whole Claude Code session, 5 Oct 2026): 59.9 M tokens (59.0 M cache reads, 0.2 M output) ≈ \$21 at API prices (Opus 5.5).

anthropic.com/news/expanding-support-for-scientists