

Engineering cumulative AI research toward the Riemann Hypothesis

Justin Sublette · Technical case study · Historical account through September 16; public-release update September 23, 2026

September 23 update: the real system is now open source

[Mathematical Research](#) is the MIT-licensed standalone research core, Mission Host and execution adapters, with fresh public starting data and its own [Linux setup](#). The private repository, research store, accounts and deployment access are not part of the release. The [engineering narrative](#) links the design choices and tradeoffs to code and observed behavior.

The [public lifecycle qualification](#) established initialization, real research-tool calls, orderly stop, restart, resumed interaction and retained state on an isolated Linux installation. The [finite-free example](#) adds a distinct successor executive that retrieved exact Evidence and Candidate revisions, retained the corrected assumptions and derived a [tighter localization bound](#) from 10 to less than 9. The public record includes selected actual requests, results and checkpoints. The baseline proof and checker were available references, so this is continued reasoning from retained work, not blind rediscovery or a measured success rate.

The [four examples](#) also include C90's scoped Loewner obstruction, a wider Jacobi construction after a restricted attempt failed, and a bounded Jensen retained-mass improvement. Their mathematical checkers and authored store walkthroughs preserve qualifications and unresolved dependencies. They do not establish an RH result or certify the larger analytic programs.

The original account below retains its September 16 horizon. The later public extraction and examples have their own recorded evidence; they do not establish that every earlier private retained-history migration or behavioral-restoration issue has been resolved. RH remains open.

The research has to outlive the conversation

I created Mathematical Research to pursue a checkable proof or disproof of the Riemann Hypothesis with autonomous AI. That is the purpose against which I judge the software. The system needs to help an agent develop mathematics, recognize useful connections, challenge its own assumptions and continue working when its current conversation ends. At the cutoff described here, RH remained unresolved, and full qualification of the newer software and restoration of autonomous scientific behavior were still unfinished.

The engineering problem is substantial even before considering the difficulty of the mathematics. A convincing argument can depend on one unjustified step. A failed construction can be remembered as if it ruled out an entire family of approaches. A useful lemma can disappear inside an unsuccessful investigation. An agent can inherit a polished summary that has lost the qualification making a result usable. Across many conversations, these errors can accumulate into a research position that looks more advanced than it is.

There is also a positive challenge: preserving enough understanding for a new agent to do something interesting. A faithful archive is valuable, but an executive needs to recognize what a result enables, where a restriction applies and why an old idea might become relevant again. The system must support that judgment while leaving room to change strategy. Continuity should carry forward what has been learned without making the current approach compulsory.

My contribution has been problem definition, system and authority design, integration, evaluation, correction and acceptance. Codex contributed extensively to implementation, investigation, mathematical exploration and

testing. The research executive is itself an AI agent: choosing mathematical strategies and directing research are part of its job. I did not design the ordinary research loop around my personally adjudicating every proof. Exact human-versus-agent line authorship was not tracked, and I do not claim to have independently derived all of the mathematical material described here.

This case study follows the decisions that shaped the system: making claims auditable, giving research a durable identity, separating strategic attention from scientific understanding, preserving work across interruptions and testing the resulting mechanisms against increasingly demanding evidence.

Making mathematical scope survive the machinery

An early audit gave the project a concrete starting point. Codex examined an inherited bundle of 23 mathematical files and concluded that it did not establish RH. The important correction was to identify where the claimed argument exceeded its support. Later algebra could still be useful, but a downstream derivation could not silently turn an unsupported prerequisite into a proved fact.

I adopted that distinction as an engineering requirement. The mathematical record needs explicit statements, dependencies, assumptions and verdicts. A researcher should be able to identify the first open dependency without reconstructing the argument from a pile of assistant-generated prose. A proposition can be registered in the authoritative record while remaining open; registration is an identity and status decision, not a declaration that the proposition is true.

The same discipline applies to negative results. There is a material difference between a counterexample to a stated claim, a failed attempt to construct something, an inconclusive search and a conditional result that remains useful. If those outcomes collapse into a single label such as “failed,” future research loses both information and options. If they collapse into “progress,” the system gives later agents premises they have not earned.

The early research campaigns explored this through separate roles for derivation, dependency review and adversarial falsification. Workers could reason from common inputs before exchanging answers. That separation was intended to reduce direct answer contamination and make disagreements informative. It did not make models from the same ecosystem equivalent to independent human mathematicians or eliminate correlated error.

One campaign provides a particularly clear example. Campaign 020 produced a proposed broad obstruction, but the class that the obstruction purported to exclude had not been specified before the search. The coordinating audit rejected the broad classification while retaining a narrower local supporting lemma. The surviving lemma did not settle RH or advance an unproved dependency by association.

The practical lesson is easy to state: showing that a chosen construction fails does not establish that every construction serving the same purpose must fail. Conversely, rejecting that larger conclusion does not erase a correct local calculation. I wanted the system to preserve both facts together, so that a successor could use the remaining mathematics without inheriting the rejected generalization.

These campaigns are part of the development history. Their old coordination machinery is no longer the current research controller. Their treatment of assumptions, counterexamples and useful partial results informed the later autonomous Mission design. Preserving scientific history did not require keeping obsolete execution engines alive.

Giving research a durable identity

The central architectural move was to separate the research objective from the agent currently pursuing it. I call the durable objective and operating envelope a **Mission**. An **executive** is the AI agent temporarily responsible for deciding what to investigate, interpreting results and maintaining the research position. Its period of authority is an executive epoch. A Mission can continue across replacement executives without treating one conversation or process as permanent.

That required several kinds of information to have distinct homes. Strategy describes present direction and why it is worth pursuing. Branches preserve particular questions and exploratory lines. Scientific Context expresses accumulated understanding. Evidence records interpreted results and their basis. Candidates preserve proposed mathematical results, including claims that might settle the target. These are revisioned research records, so an exact earlier statement remains identifiable when later thinking changes.

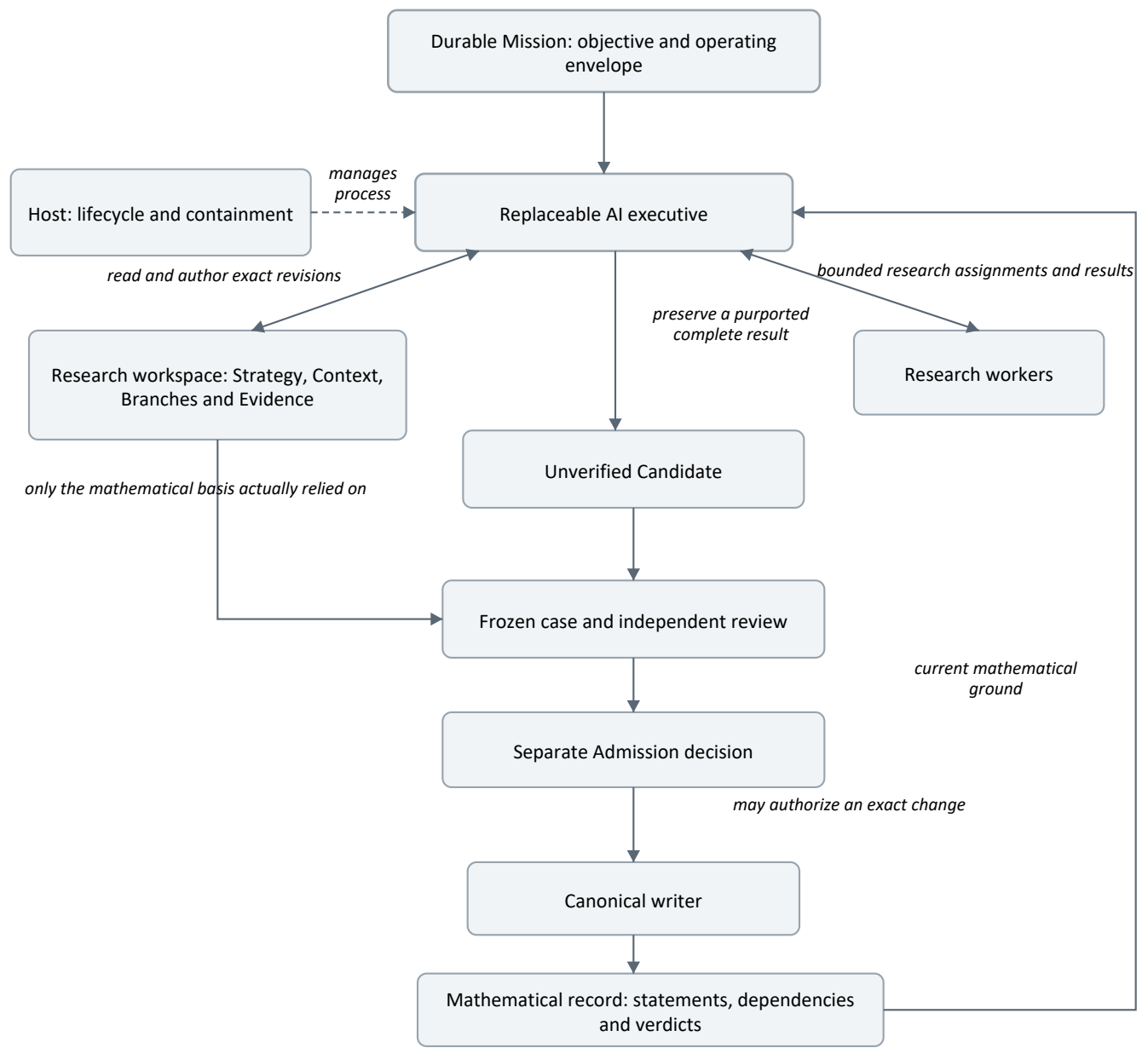
The distinction matters during ordinary research. A new result can change the best next investigation without changing the objective. A branch can become unpromising while leaving behind a useful technique. Strategy can move elsewhere without deleting that technique. A successor needs to recover those relationships, then remain free to revise the inherited strategy.

Checkpoints serve that continuity. They carry compact, exact anchors into already-authored research state. They are useful because the meaning lives in the records they identify. Replacing the executive should not require an outgoing agent to compress the entire scientific history into one final message, nor should the checkpoint itself become another competing scientific record.

The implemented workspace uses a single SQLite writer, immutable revision identities and content-addressed material. The writer provides one place to apply authorized state changes. Revision identities distinguish the result an agent actually saw from a later version. Content addressing lets the system check whether retrieved material matches the bytes identified by its recorded digest. These mechanisms protect identity, consistency and provenance; they cannot establish that an argument in those bytes is mathematically valid.

The Mission Host has a different job. It manages process lifecycle, containment and recovery. Workstation Control owns concrete process and provider attempts and their raw operational evidence. Those responsibilities support the research executive but do not give the Host authority to choose mathematics. A service restart cannot rewrite Strategy or convert a failed attempt into a scientific result.

The diagram shows these responsibilities and the separate route for a purported complete result. It is an architecture view, not a claim that every path had completed live qualification by the cutoff.



Recovery made the separation concrete. A dead process did not necessarily mean that its durable authority had been reconciled. In the recorded recovery lineage, fencing a failed epoch could leave a Mission revoked. Recovering its state, granting renewed authority, revising model policy and starting another executive therefore became separate operations. Earlier attempts retained their original policy and failure history. A new revision could authorize future work without editing the past into success.

That is the kind of continuity I was designing for: a new agent can recover a truthful research position, including what remains unresolved, and then exercise fresh judgment within an explicit operating envelope.

Scientific understanding needs more than an archive

Context became one of the most consequential design problems. An earlier opening reconstructed a broad view of Mission history automatically. As the retained record grew, that reconstruction became an unsuitable default input. The September 2 design replaced it with deliberate orientation and exact retrieval: the executive received current purpose, Strategy and selected ground, while fuller history remained accessible when needed.

This addressed one problem but left another. Current Strategy describes where attention is being allocated. It does not necessarily express everything scientifically important that the Mission has learned. If automatic exposure depends entirely on current Strategy, an agent may be well informed about today's plan and poorly positioned to recognize an unexpected connection elsewhere.

The September 14 change gave cumulative scientific Context an identity independent of Strategy. Its purpose is to maintain an intelligible account of questions, capabilities, restrictions and relationships that remain relevant to the research. That account evolves through correction, replacement and synthesis. It is not a chronological entry for every output or a repeated summary of the whole archive.

For example, understanding a result may require knowing the question it helps answer, the assumptions under which it applies and which broader conclusions it does not support. Those qualifications belong with its treatment. If a successor sees the useful conclusion without its restriction, retrieval has preserved the words while damaging their meaning.

The exposure contract therefore carries a root treatment and explicitly selected deeper treatments, with relevant enclosing qualifications. It avoids recursively expanding every backing source into the opening. Exact history can still be retrieved. The choice of what to expose is deliberate, but the design does not promise that arbitrarily growing scientific knowledge can always be represented faithfully at constant cost.

I also needed to distinguish several uses of a reference. An old result might help the executive recognize an opportunity. It might supply background about how an approach developed. Or a proposed proof might actually depend on its mathematical content. Those are different relationships. Merely seeing a current Context treatment does not make every fact behind it a premise. When a Candidate does rely on material, its basis must identify the exact relevant revision and treatment rather than silently substituting a newer head.

This separation supports both discovery and review. Discovery can range across historical connections without expanding every Candidate's proof obligations to the entire workspace. Review can inspect what the argument actually uses without mistaking a background reference for an asserted theorem dependency.

The system also needs to look beyond already-recognized connections. A material new capability, a scoped failure or a changed qualification can affect other parts of the research even before an agent suspects which old result matters. The executive is expected to consider those consequences and may investigate directly or assign temporary historical discovery work. An inquiry about a known relationship and a search for an unanticipated connection require different approaches.

This does not create a standing historian that must process every result before research continues. A consequential decision with a material historical dependency needs appropriate grounding; unrelated work does not automatically wait. The intent is to improve research judgment while keeping the supporting mechanism proportional to the decision it serves.

At the storage boundary, the newer implementation added authenticated indexed discovery over declared owner-content fields. Searches can refer to a current or historical cut of the records and return bounded results through the authorized interface. This supports targeted access without making every query scan all raw material. Missing required index state is reported explicitly. The architecture establishes a route for finding and checking material; it does not establish that an agent will notice every useful connection.

The resulting distinction is central to the system: the archive preserves what happened, Strategy allocates attention, and scientific Context carries evolving understanding. They work together while answering different questions. By the cutoff, the new Context, publication and indexing mechanisms existed in source and had focused verification results. Improved autonomous scientific behavior remained to be demonstrated.

The restoration goal concerned specific research behaviors. Earlier retained traces showed examples of preserving a parent program after correcting one component and recombining useful ingredients across branches. The newer design sought to recover those capabilities. Those examples did not establish a fully qualified earlier autonomous system, and missing historical startup and execution details limited causal comparison. Restoration therefore remained a behavioral objective, not a measured regression-and-recovery result.

Preserving results when execution is interrupted

Research continuity also depends on what happens between deciding to save something and learning that the save succeeded. An agent might produce an Evidence result and an already-authored continuation explaining the unfinished science that follows from it. If those two writes occur separately, an interruption can leave the result preserved but lose the paired continuation needed for the intended exposure.

The newer publication route can commit one explicitly paired Evidence result, its authored scientific treatment and the necessary exposure through the existing Store writer. That is a narrow atomic operation. It does not make an entire synthesis or all related research one transaction. Independent sibling work and immediate preservation of a potential complete result remain separate, so failure of the pair does not erase them.

Acknowledgement loss requires a related distinction. If the writer committed a request but the caller did not receive the response, a retry should recover the original effect. It should not require the model to repeat completed scientific work or accidentally create a different result. Exact-request replay binds recovery to the committed request and its original state relationships.

These details are consequential because the system is managing authored knowledge across unreliable process boundaries. A record's existence, the caller's knowledge of that existence and the next agent's ability to recover its meaning are separate facts. The design aims to preserve their relationship explicitly.

The implementation needed corrections in publication schemas, source authentication and replay behavior. Focused tests subsequently exercised repaired boundaries, while broader structural runs still contained failures or unfinished cases. Those passing subsets support their specific mechanisms. They do not turn the unfinished aggregate into a qualified whole.

A complete-result claim must be preserved before it can be accepted

The research loop must handle an especially important case: an agent believes it has a complete proof or disproof. My design requires immediate preservation of that purported result as an unverified Candidate. Its significance or incomplete review preparation should not cause the claim to be discarded before it can be examined.

The complete-result attention path is called A1. It redirects broad exploration toward reconstructing, testing or falsifying the exact claim. A1 does not certify the result. A concrete mathematical defect can invalidate the particular claim while preserving its history and any surviving useful work. Generic uncertainty or difficulty arranging review is not itself such a defect.

Preserving the Candidate is therefore distinct from admitting a change to the mathematical record. Admission freezes the exact Candidate and the mathematical basis actually relied upon. An independent reviewer examines that case. A separate decision role determines whether an exact change is warranted, and a

dedicated canonical writer applies the authorized change. The Candidate author, reviewer and admitting role are distinct for that case.

Here, independence means separate authorship and judgment roles examining the exact case. The reviewer is expected to reconstruct critical steps and challenge assumptions, circularity and composition; fresh-context rederivation is preferred. This does not claim external human review or guarantee freedom from shared model errors. The case must survive substantive scrutiny rather than gain authority from a count of agreeing agents.

Those roles are part of the autonomous research design; they do not imply that I personally serve as the routine mathematical reviewer or that an outside tribunal must participate before internal research can continue. They also do not make agreement a proof. The argument must still support its exact conclusion, and the software must preserve the distinction between evidence, a decision and the successful canonical transaction.

This is deliberately compatible with revising the research approach. A failed route can be closed at its actual scope, while another route to RH remains available. In the project's existing theta-to-Stieltjes approach, the source bridge and both unshifted and shifted Hankel obligations remain open. These are obligations of that particular mathematical program. A genuinely different program must establish its own complete, noncircular implication to RH rather than inherit unrelated route-specific machinery.

The current source contains the separate Admission path, but the inspected evidence does not establish live use of that path or a resulting canonical mathematical change. Public disclosure is a further owner decision. No model output, successful checkpoint or green software suite crosses those boundaries by itself.

Establishing what the implementation can actually do

One of the strongest positive execution results is the reproducibility of a frozen finite mathematical-software suite. At one exact pushed revision, three fresh remote replicas ran sequentially, each taking approximately 104 minutes. Each completed the prescribed sequence of software checks, numerical diagnostics and exact computation. The normalized mathematical-result streams matched within and across the replicas, collection was complete and the canonical proof state remained unchanged.

Getting there exposed concrete environment problems. A read-only source root affected lock placement. Host confinement affected dependency visibility. Python import behavior depended on assumptions that did not hold under the intended execution conditions. Correcting those boundaries made the frozen procedure reproducible while preserving its containment.

That is a meaningful software result: another fresh execution of the specified finite procedure produced the same semantic output under the recorded conditions. It is not an independent proof of RH or an argument from finite checks to an infinite theorem. The distinction lets the execution result retain its full value without acquiring a different mathematical meaning.

Later work tested continuity with retained research data. The recorded isolated Linux staging rehearsal reconstructed a workspace, performed a write, committed and read a checkpoint, and reached a stopped successor boundary with zero model-provider calls. This exercised a bounded persistence and succession path. Because no model was called, it did not show a successor conducting creative mathematical research.

Another development established a shared immutable build artifact. Three selected Host/bridge tests used the same exported build bytes without rebuilding them or selecting a research release. That helps separate a result about a specific artifact from a result about whatever happened to be rebuilt locally. These tests used substituted model/configuration inputs and a simulated installation layout. They were selected integration checks, not full qualification of the later software.

The retained historical record provides stronger primary execution evidence for the earlier finite replicas. The newer staging and selected native results are supported here by committed, revision-specific execution accounts; their primary run artifacts were not independently recovered for this case study. That difference belongs with those results. Neither set supports a claim of new mathematical discovery merely because the software ran successfully.

I treat these observations as evidence about distinct properties: reproducible computation, persistent research state, checkpoint succession and consistent build identity. They make the system more concrete and reviewable. They leave the separate question of sustained scientific capability open.

What authentic history exposed

The next stronger test was to prepare a disposable derivative of authenticated retained research data under the newer storage schema, then read actual Evidence and compare its linked source bytes with authenticated originals. Synthetic migrated examples had passed, but authentic history contained older relationships that the newer readers needed to interpret faithfully.

Preparation exposed historical provenance problems and an unidentified Mission-scope failure. Separate diagnosis reproduced a defect in how older Branch records expressed Mission identity; the failing scope check itself had not revealed which kind of record triggered it. The repair followed authenticated historical relationships and checked their consistency. It preserved the original bytes and their actual meaning instead of inventing modern execution requests for old material or substituting a current revision for the historical one.

The failures also prompted examination of redundant preparation work. Index construction moved from a large in-memory posting structure toward scratch SQLite ingestion, and later corrections reused canonical posting bytes and removed specific redundant audit work while preserving required checks. Focused native regressions passed for these changes.

The authentic preparation nevertheless remained unsuccessful within the cutoff. Six recorded attempts failed before the intended Evidence/source comparison; three later attempts ran for approximately an hour. For those three later failures, the returned records did not retain the terminal reason or final internal phase. The evidence did not justify attributing them to out-of-memory termination or claiming a measured whole-corpus speedup from the repairs.

The final changes in this interval improved bounded diagnostic observations so that a later failure could retain more useful information. They established implementation and a recorded controller update, not the outcome of a subsequent successful run. Source-preservation, stopped-state and selection checks had their own successful results, which remained distinct from completing the preparation.

This episode matters because a durable research system must remain able to interpret the history it actually accumulated. Passing clean fixtures is useful; authentic retained data can reveal assumptions those fixtures never exercised. The resulting corrections improved specific readers and preparation mechanisms, while the unfinished test continued to identify a real engineering frontier.

The contribution and the remaining frontier

The strongest account of this work is the connection between its research purpose and its technical decisions. I defined and integrated a system in which an autonomous executive can change strategy, preserve useful mathematics, recover from interruption and submit a purported complete result to a separate scrutiny path. Extensive AI collaboration was how much of that design, implementation and investigation was carried through. My role included deciding which distinctions the system must preserve and accepting corrections when its machinery failed to preserve them.

The implemented foundation spans Python, TypeScript and Node.js; SQLite and content-addressed material; cryptographic commitments and exact revision identities; and a Linux Host with controlled remote execution. Those technologies matter through the jobs they perform: keeping authorship and evidence attached to exact research, preventing ambiguous execution from silently becoming another attempt, and allowing a successor to recover a meaningful position.

As of September 16, the project had established substantial software mechanisms and bounded execution results. Full qualification of the newer build, successful completion of the authentic retained-data preparation and restoration of autonomous scientific behavior remained unfinished. The open RH obligations remained open.

The continuing standard is whether the next researcher can recover what is known, understand what a result actually permits, see where the argument stops and pursue a promising next question. A correct, independently checkable proof or disproof remains the intended outcome. The software earns its place by helping that work proceed faithfully and cumulatively.