

RESEARCH PROGRAMME GUIDE · R01

Dweve research: current questions, prototypes, and the path to product

Jacquard, Forge, and Mycelia, what each asks, how evidence is evaluated, where status stops, and what promotion would require.

RESEARCH

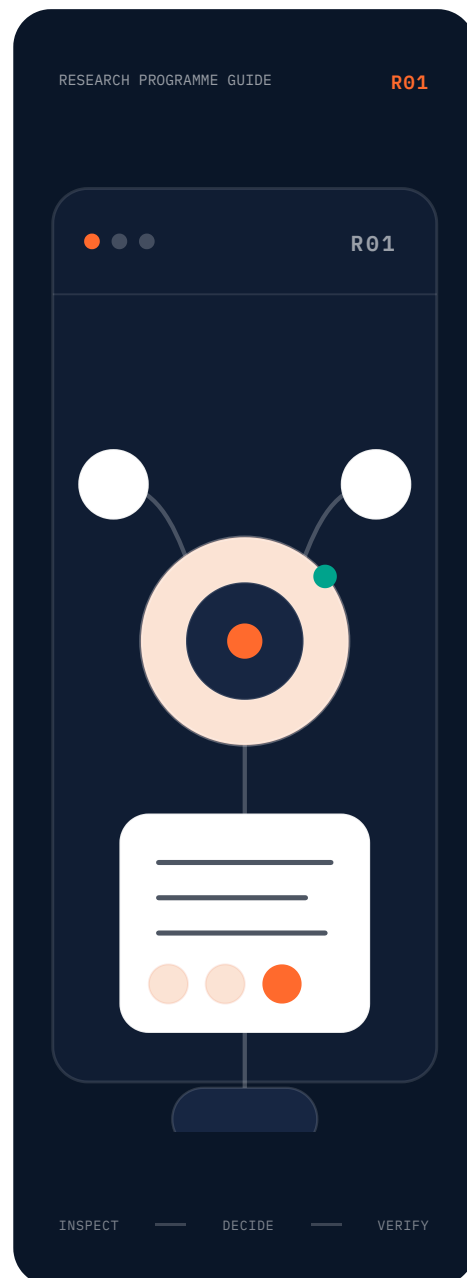
STATUS

REPRODUCTION

FAILURE

PARTNERSHIP

PROMOTION



Read this when you need to separate an ambitious research programme from a current product promise, and evaluate both honestly.

R01

CONTENTS

From research question to an explicit readiness decision

Three current research lines are explained through mechanism, artefacts, evidence, status, failure and partnership; the final section defines what product language would require.

PRIMARY READERS

Researchers, engineering partners and product reviewers

READING DEPTH

Research + technical

DECISION THIS SUPPORTS

A status-disciplined research map for partners and advanced evaluators, without converting prototypes or measurements into present commercial capability.

01

Read this first

03–05

Audience, decision, claim discipline, status vocabulary, and programme map

02

Current research questions

06–19

Publication boundary, research contract, lineage, illustrative promotion scenario, and three three-page research-line records

03

How research is evaluated

20–27

Reproduction, benchmarks, negative results, data/ethics, packets, security/operations, and partnership

04

The path to product

28–35

Contract, correctness, security, performance, documentation, deployment/maintenance, and readiness board

05

Closing research decision

39

Choose a bounded question, preserve the evidence, and record the next status

06

Subject index

36–38

An alphabetical finder for concepts, controls, products, and decisions.

STATUS RULE

Every mechanism and number remains attached to its published website sources, version, method and status. Research access is not product availability.

SOURCE DISCIPLINE

Dweve-specific facts are taken directly from the English website text audited 2026-07-20; web-navigation wording is humanised for print. Evaluation guidance is editorial, and scenarios and derived visuals are illustrative. Unsupported synthesis is replaced by canonical copy or omitted.

READ THIS FIRST • READER DECISION

Who should read this programme guide

Research organisations, technical partners, advanced customers, funders, candidates, and engineers evaluating Dweve's longer-term technical direction.

Use this guide to understand which research lines the website presents, why each exists, how it relates to current products or open foundations, what evidence is described, and where research must not be mistaken for shipping or contracted capability.

The security surface is not a questionnaire you answer later, it is a set of product properties an engineer can read off the substrate. Data residency in Europe, bit-identical replay, an Ed25519-signed Knot ledger with a per-run Merkle root, GDPR by design, and explainable output from discrete binary constraints. Knot is open source on github.com/dweve-ai under Apache 2.0.

The website is the only factual source for this edition. Methods and readiness procedures in this guide are evaluation requirements and illustrations. Verify current repositories, packages, builds, partner access, evidence, IP/licence, release status and commercial terms at the decision date.

PROGRAMME**Three routes**

Jacquard, Forge, Mycelia

DECISION**Status + fit**

Research question, evidence, partnership, promotion

REQUIRED RESULT**Research packet**

Question, lineage, method, raw result, limits

RED LINE**No status inflation**

Research is not present product capability

READER DECISION

Understand the current research lines, evidence and boundaries; decide whether one warrants a scoped research partnership or only continued observation.

READ THIS FIRST • STATUS VOCABULARY

Status words have different operational consequences

Use a status only when the current published description, artefact, owner, evidence and access terms substantiate it.

Question, hypothesis, and prototype

A question identifies uncertainty. A hypothesis predicts an observation. A prototype exercises a proposed mechanism. None promises release, maintenance, security, target support, availability or general validity.

Benchmark snapshot and reproduced result

A snapshot records a measurement under one method. Reproduction repeats it under the same scope. Replication challenges it with independent implementation, data, investigators or environment. Each retains negative results and limitations.

Open foundation and supported product

You open Fabric in the browser and pay for the first month. After that month, an eligible customer may choose the published spare-compute option. Dweve shows the current eligibility, contribution requirements, controls, data boundaries, and exit terms before you opt in.

STATUS	MINIMUM EVIDENCE
Research question	published website sources + owner + intended investigation
Prototype	Runnable artefact + version + method + limits
Measured snapshot	Hardware/corpus/config + raw result + date
Reproduced	Independent run + comparison + divergence
Open foundation	Public/current source + licence + package/status
Supported product	Delivered scope + owner + assurance + docs + terms

A project can occupy more than one status only for clearly separated artefacts and scopes.

AVAILABILITY RULE

Technical detail and partner access do not by themselves establish public availability, source rights, production support, or a contract commitment.

READ THIS FIRST • CANONICAL SPOTLIGHT

The current programme map

Jacquard, Forge and Mycelia are the three current research routes.

Jacquard studies a browser-native, typed and replayable engine for agent-built interactive worlds. Forge studies synthesis and solver-backed verification of optimised low-level programmes. Mycelia studies bottom-up adaptive intelligence in a deterministic, conservation-audited simulated world.

The lines share a preference for explicit contracts, reproducible artefacts and platform relevance, but their scientific and engineering questions are not interchangeable. A typed API does not answer a synthesis proof; a token benchmark does not establish artificial-life emergence; deterministic replay has different scope in a game engine and substrate simulation.

Reproducibility starts before execution. We model the question as a falsifiable contract, preserve provenance at collection, identify every transformation and runtime condition, and connect observations to downstream claims. A replay should either reproduce the path or expose the condition that moved.

LINE	QUESTION	CURRENT BOUNDARY
Jacquard	Agent-built interactive worlds through typed tools	Research; release terms to check
Forge	Synthesised kernels with verification and target codegen	Research partner access
Mycelia	Emergence under deterministic substrate constraints	Research partner access

Turns a recurring source question into typed records with the collection route attached.

PROGRAMME PRINCIPLE

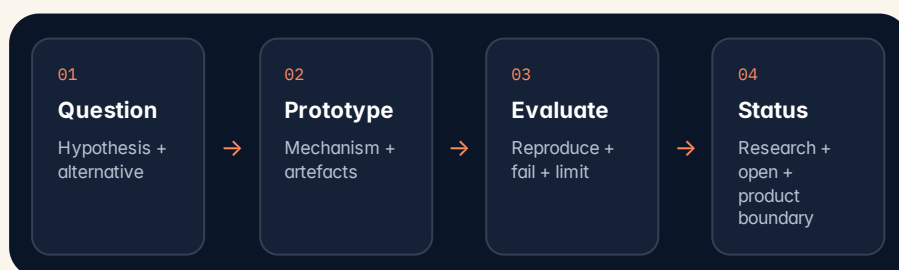
Question → method → raw evidence → failure and limits → reviewed claim → explicit status.
Product language requires a separate gate.

01

CURRENT RESEARCH QUESTIONS

Keep question, mechanism, evidence, and status connected

The first section establishes the programme discipline, then gives Jacquard, Forge, and Mycelia three complete pages each.



The programme does not skip from an interesting prototype to product language.

PROGRAMME • WHY SEPARATE IT

Research needs its own publication boundary

Future work can be ambitious and specific without being presented as a present product, open-foundation release, support commitment, or commercial entitlement.

The current website publishes research questions and technical detail because inspectability begins before a result becomes a product. A separate research programme lets partners understand the hypothesis, mechanism, artefacts and evidence while keeping product and contract language honest.

Label question, hypothesis, design, prototype, experimental branch, benchmark snapshot, reproduced result, open foundation, product integration, supported release and roadmap separately. Status belongs to a named published source, version and date. A strong benchmark or complete-looking interface does not advance a project automatically.

Commercial and product guides may reference research to explain direction or provenance, but they may not list research capability as available unless the current product description, delivered manifest and written terms establish it. A separate open-source component must remain in its own catalogue rather than being recast as a research line.

PUBLICATION	ANSWERS	DOES NOT PROMISE
Research website description	Question, mechanism, experiment, evidence	Supported availability
Benchmark	Result under one method	General performance or SLA
Open foundation	Current source/package/licence status	Commercial product/support
product description	Owned responsibility and current proposition	Every roadmap item
Delivered manifest	What this version actually contains	Future compatibility
Agreement	Rights, service, support and remedies	Scientific or technical truth alone

The publication type is part of claim meaning.

RED LINE

Never convert “the website describes a research track” into “the customer receives this product capability” without current delivery and contract evidence.

PROGRAMME • QUESTION

Begin with a falsifiable research contract

The unit of enquiry, expected observation, acceptance rule, and kill condition are fixed before a run can move the answer.

Write the problem and why it matters, unit of enquiry, hypothesis and serious alternative, expected observation, inputs and rights, method and implementation, environment, comparison or oracle, metric, acceptance and rejection rule, sample/repetition, uncertainty, confounders, resource budget, safety/ethics, raw evidence, reviewer and change policy.

Preregister where the research question would otherwise move after seeing output. Version amendments and explain why they were made. Exploratory work remains valuable when labelled exploratory; it cannot be reported as a confirmatory result against a hypothesis defined after the observation.

Define kill conditions for invalid method, rights or ethics failure, security boundary breach, conservation or correctness failure, unrecoverable divergence, resource ceiling, unavailable independent comparison, or a result whose operational harm exceeds the programme's authority. Stopping is a valid research outcome.

ILLUSTRATIVE RESEARCH CONTRACT

study_id	dweve-r-4f9e
question	one falsifiable proposition
hypothesis	expected observation
alternative	serious contrary result
inputs	identities + versions + rights
method	implementation + configuration
environment	hardware + runtime + dependencies
comparison	oracle/baseline + acceptance rule
kill	invalid / unsafe / divergent / over-budget
evidence	raw results + lineage + review

Illustrative fields. Use the partner's approved research and ethics process.

MUTATION TEST

Change the hypothesis, corpus, seed, method, environment, metric, or acceptance rule after the run. The new result must carry a new contract version and relationship to the earlier question.

PROGRAMME • LINEAGE

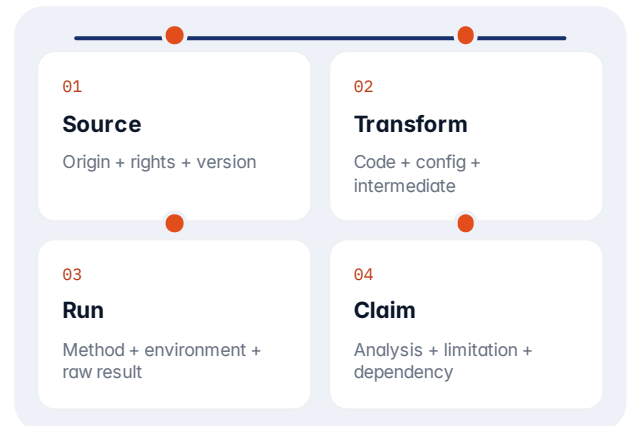
Provenance and transformation belong in the graph immediately

A result inherits every source, cleaning step, generator, build, configuration, environment and analysis decision behind it.

Identify source and authority, collection context and time, rights/consent/ethics, raw object hash or protected reference, schema and quality, transformations and code, intermediate objects, exclusions, model or simulator inputs, configuration, software/build/runtime, hardware target, random seed, observations, analysis, figure/table and conclusion.

Store direction both ways: which inputs and transformations produced this claim, and which claims, figures, models, decisions or products depend on this source or component. When a source is corrected, rights change, or an implementation is found faulty, the graph should identify affected work without erasing the earlier historical state.

Sensitive or licensed material can remain in an authorised repository. The research packet carries resolvable identity, access and retention policy, integrity and the procedure needed to reproduce or independently review within permission. Public reproducibility is not the same as unrestricted data publication.



Research lineage is a queryable graph rather than a citation list assembled at publication time.

CORRECTION TEST

Correct one upstream object and require the graph to identify affected runs and claims, preserve the old version, trigger scoped review, and publish an explicit old-to-new transition.

PROGRAMME • REQUIRED SCENARIO

The worked promotion scenario is illustrative at every step

Follow one research result from hypothesis to prototype, reproduction, failure review, contract, and a hypothetical product-readiness gate without turning the scenario into a roadmap promise.

Illustration: a Forge partner proposes synthesising a bounded integer kernel used repeatedly in a Core-like runtime. The hypothesis predicts equivalent output under declared preconditions with lower latency on one target. A reference and solver obligation are pinned; search produces candidates; counterexamples refine the specification; a final candidate is compiled and benchmarked.

Independent reproduction finds parity on the original target but divergence on a second target because an overflow precondition was incomplete. The result does not “almost pass”. The specification is corrected under a new version, the earlier candidate remains failed for the broader domain, and search and verification repeat. The narrower finding may still be useful.

A hypothetical promotion gate then asks for a stable operation contract, supported domain, reference and property corpus, solver/proof artefacts, all target implementations, security and undefined-behaviour review, performance evidence, documentation, packaging, maintenance owner, compatibility and rollback. Passing would authorise a separate product release decision, not make this illustration real.

01

Hypothesis

Bounded equivalence + performance prediction.

02

Prototype

Candidate + search trace + proof obligation.

03

Reproduce

Independent target, corpus and raw evidence.

04

Fail/revise

Overflow counterexample changes the specification.

05

Contract

Stable types, domain, errors, artefacts and tests.

06

Readiness gate

Security, targets, performance, docs, owner, support.

ILLUSTRATIVE ONLY

No part of this scenario is a claim that the named kernel, candidate or promotion exists. It demonstrates how research status should change under evidence.

PROGRAMME LINE • JQ

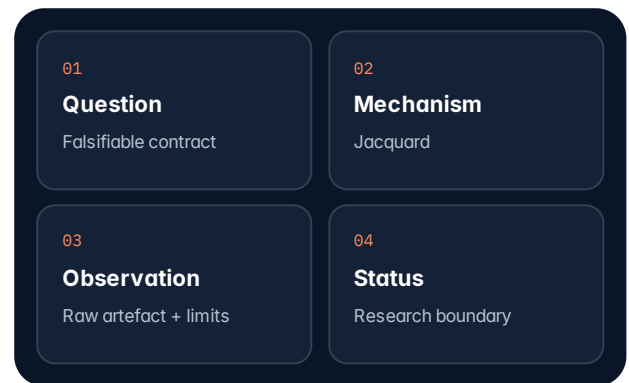
Jacquard: the research question and proposed mechanism

Can an AI agent build and change a complete interactive world through the same typed operations a human editor uses, while every input and state transition remains replayable?

Can an AI agent build and change a complete interactive world through the same typed operations a human editor uses, while every input and state transition remains replayable?

Jacquard is described as a web-native game engine for agent-built worlds. Terrain, scenes, entities, components, assets, materials, physics, lighting, audio, UI, quests, NPCs, build, debug and playtest are exposed through typed tools. The browser editor is a non-privileged client of the same surface, and an MCP server connects agent platforms.

The research boundary is: Aura, Nexus and Fabric are described as potential callers of the same typed API. Jacquard should not be listed as an available product or open foundation unless the current published website sources and delivered terms establish that status. A diagram or functioning prototype can make the mechanism concrete; it cannot establish generality, product readiness, support, security, performance, scientific validity or commercial availability without the corresponding evidence and ownership.



The minimum Jacquard research path.

PRIMARY QUESTION

Can an AI agent build and change a complete interactive world through the same typed operations a human editor uses, while every input and state transition remains replayable?

PROGRAMME LINE • JQ

Jacquard: artefacts, measurements, and reproduction

A research claim becomes inspectable when inputs, method, environment, raw results, comparison, failures, and limitations retain stable identities.

The material research artefacts are world/project identity, typed operation and parameters, asset and component state, event log, input timestamps, fixed-timestep and seeded-RNG configuration, build or portable bundle, playtest result, error state, and replay comparison.

The published claims deterministic-by-default execution using a fixed timestep, seeded random generator and timestamped input recording. It describes replay from seed and input log so a bug report can reproduce the frames observed. These statements need a pinned build, browser/runtime, asset set, seed, log and comparison.

Measure browser and runtime compatibility, tool coverage and semantic parity between editor and agent surfaces, deterministic replay, bundle portability, asset fidelity, resource use, build/playtest failures, and security/capability behaviour. Capture corpus or task and rights, hardware and topology, software/build and dependencies, configuration and seeds, warm/cold and concurrency, duration/repetitions, metric and comparison, raw output, failures, reviewer and date. A different question or environment can produce a different result without falsifying a narrower published measurement.

RESEARCH RECORD		jacquard-reproduction
QUESTION	unit + hypothesis + acceptance	FROZEN
INPUT	identity + rights + versions	PINNED
METHOD	implementation + config + environment	PINNED
OBSERVATION	raw output + failure + resources	RETAINED
COMPARISON	baseline/oracle + rule	DECLARED
VERDICT	supported / limited / negative / divergent	REVIEWED
A screenshot, headline number or successful demo is not a reproduction record.		

MEASUREMENT BOUNDARY

browser and runtime compatibility, tool coverage and semantic parity between editor and agent surfaces, deterministic replay, bundle portability, asset fidelity, resource use, build/playtest failures, and security/capability behaviour

PROGRAMME LINE • JQ

Jacquard: status, failure, and a credible partnership

Research engagement begins with a bounded question and makes release, IP, publication, data, infrastructure, support, and promotion assumptions explicit.

the current published description repeatedly labels Jacquard a Dweve research track or research project and says it is not a public open-source release today. It describes browser-native evaluation and instructs readers to check public release terms when announced.

The principal failure model includes Tool or editor semantic drift, non-deterministic frame/state, missing asset, incompatible browser/runtime, unbounded capability, malformed operation, bundle loss, replay mismatch, or a release/right assumption that the current published description does not support. Preserve counterexamples, inconclusive runs, changed hypotheses, reproduction failures and unsupported environments. A negative result narrows the search space; hiding it weakens the programme and can make a later product unsafe.

Partner design: A suitable evaluation supplies a bounded lesson, training simulation, prototype or embedded interactive scenario, accepted assets and behaviours, browser/security boundary, deterministic test cases, and a clear rule for publication or private retention. Agree purpose, people, data and code rights, infrastructure and jurisdiction, security, access, methods, raw evidence, publication and review, IP and open-source intentions, cost and contribution, support, exit, and who may make current availability claims.

STATUS QUESTION	REQUIRED ANSWER
What exists?	Code/prototype/package, version, artefacts, owner
Who can access?	Public, partner, on request, private; actual terms
What is measured?	Task, hardware, method, raw evidence, date
What is supported?	Platforms, documentation, maintenance, security, service
What may be published?	Data/code/result/IP open-foundations overview agreement
What would promote?	Contract, tests, assurance, operation and owner gates

Jacquard status is decided from current evidence and terms, not the maturity implied by its technical detail.

FAILURE TO INJECT

Tool or editor semantic drift, non-deterministic frame/state, missing asset, incompatible browser/runtime, unbounded capability, malformed operation, bundle loss, replay mismatch, or a release/right assumption that the current published description does not support.

PROGRAMME LINE • FG

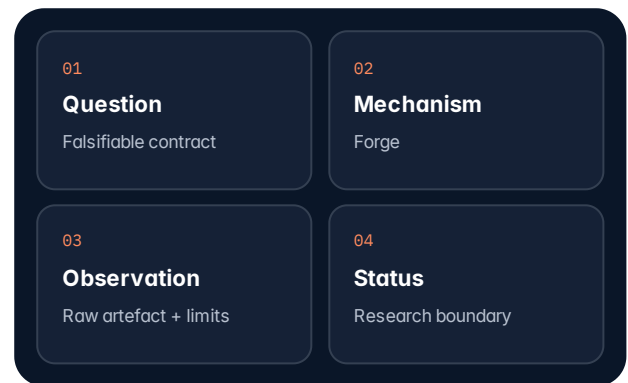
Forge: the research question and proposed mechanism

Can program synthesis discover a faster or smaller implementation for a precisely specified kernel and prove equivalence strongly enough for a human to consider using it?

Can program synthesis discover a faster or smaller implementation for a precisely specified kernel and prove equivalence strongly enough for a human to consider using it?

Forge is described as a program-synthesis engine over typed expressions with enumerative, CEGIS, genetic-programming and MCTS search; e-graph equality saturation; Cranelift JIT measurement; Z3 and CVC5 encodings; ML-guided cost models; and multi-objective Pareto search over latency, operation count, memory and register pressure.

The research boundary is: A successful candidate could inform Core operations, Kera lowering or another documented runtime, but promotion requires a stable contract, target matrix, verification corpus, security review, maintained implementation, documentation and release ownership. A diagram or functioning prototype can make the mechanism concrete; it cannot establish generality, product readiness, support, security, performance, scientific validity or commercial availability without the corresponding evidence and ownership.



The minimum Forge research path.

PRIMARY QUESTION

Can program synthesis discover a faster or smaller implementation for a precisely specified kernel and prove equivalence strongly enough for a human to consider using it?

PROGRAMME LINE • FG

Forge: artefacts, measurements, and reproduction

A research claim becomes inspectable when inputs, method, environment, raw results, comparison, failures, and limitations retain stable identities.

The material research artefacts are kernel specification and types, preconditions and defined domain, reference implementation, candidate programme/IR, search engine and configuration, search trace and budget, proof obligation and solver result, counterexample corpus, compiled target artefact, performance results, and Pareto record.

The route currently names 27 pre-configured synthesis tasks, more than 3,500 tests, 3–12× multi-core parallel-synthesis speedups, ML-guided 100–1000× search acceleration, and targets including x86-64, RISC-V, WASM, Vulkan GPU and Verilog. Each figure remains a published claims tied to its actual task and method.

Measure proof coverage and assumptions, counterexamples, search completeness within budget, candidate equivalence, target correctness, compile validity, latency distribution, memory/register use, operation count, search time and reproducibility. Capture corpus or task and rights, hardware and topology, software/build and dependencies, configuration and seeds, warm/cold and concurrency, duration/repetitions, metric and comparison, raw output, failures, reviewer and date. A different question or environment can produce a different result without falsifying a narrower published measurement.

RESEARCH RECORD		forge-reproduction
QUESTION	unit + hypothesis + acceptance	FROZEN
INPUT	identity + rights + versions	PINNED
METHOD	implementation + config + environment	PINNED
OBSERVATION	raw output + failure + resources	RETAINED
COMPARISON	baseline/oracle + rule	DECLARED
VERDICT	supported / limited / negative / divergent	REVIEWED
A screenshot, headline number or successful demo is not a reproduction record.		

MEASUREMENT BOUNDARY

proof coverage and assumptions, counterexamples, search completeness within budget, candidate equivalence, target correctness, compile validity, latency distribution, memory/register use, operation count, search time and reproducibility

PROGRAMME LINE • FG

Forge: status, failure, and a credible partnership

Research engagement begins with a bounded question and makes release, IP, publication, data, infrastructure, support, and promotion assumptions explicit.

Forge is presented as available through Dweve's research partner programme. Research access is not the same as a supported product, general licence, release commitment, or guarantee that a generated kernel is suitable for production.

The principal failure model includes Under-specified behaviour, solver encoding gap, undefined or overflow domain, proof timeout, counterexample after apparent success, target/codegen mismatch, benchmark gaming, non-reproducible search, unsafe generated construct, or no maintained product owner. Preserve counterexamples, inconclusive runs, changed hypotheses, reproduction failures and unsupported environments. A negative result narrows the search space; hiding it weakens the programme and can make a later product unsafe.

Partner design: A useful partner contributes a small high-value kernel with a formal or testable specification, legal input rights, relevant target hardware, current baseline, correctness oracle, performance objective and an engineer able to review counterexamples and generated code. Agree purpose, people, data and code rights, infrastructure and jurisdiction, security, access, methods, raw evidence, publication and review, IP and open-source intentions, cost and contribution, support, exit, and who may make current availability claims.

STATUS QUESTION	REQUIRED ANSWER
What exists?	Code/prototype/package, version, artefacts, owner
Who can access?	Public, partner, on request, private; actual terms
What is measured?	Task, hardware, method, raw evidence, date
What is supported?	Platforms, documentation, maintenance, security, service
What may be published?	Data/code/result/IP open-foundations overview agreement
What would promote?	Contract, tests, assurance, operation and owner gates

Forge status is decided from current evidence and terms, not the maturity implied by its technical detail.

FAILURE TO INJECT

Under-specified behaviour, solver encoding gap, undefined or overflow domain, proof timeout, counterexample after apparent success, target/codegen mismatch, benchmark gaming, non-reproducible search, unsafe generated construct, or no maintained product owner.

PROGRAMME LINE • MY

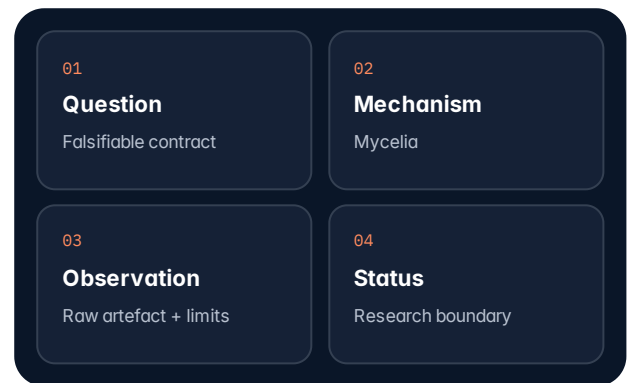
Mycelia: the research question and proposed mechanism

Can adaptive behaviour emerge in a deterministic world with explicit physics, chemistry, biology, ecology and information constraints, without a human-defined reward function?

Can adaptive behaviour emerge in a deterministic world with explicit physics, chemistry, biology, ecology and information constraints, without a human-defined reward function?

Mycelia is described as an artificial-life substrate with discrete time and 15 phases per tick, six metabolic pools, seven diffusing chemicals, an 88-byte genome and a stack-based virtual machine with 73 instructions. Cells survive under an energy budget; physics, chemistry, biology, ecology and information engines run together.

The research boundary is: Numerus-style deterministic arithmetic, Core/Kera targets, event/evidence foundations and Twin/FMI concepts may inform or support the research architecture. Those relationships must be demonstrated in the current build rather than assumed from shared vocabulary. A diagram or functioning prototype can make the mechanism concrete; it cannot establish generality, product readiness, support, security, performance, scientific validity or commercial availability without the corresponding evidence and ownership.



The minimum Mycelia research path.

PRIMARY QUESTION

Can adaptive behaviour emerge in a deterministic world with explicit physics, chemistry, biology, ecology and information constraints, without a human-defined reward function?

PROGRAMME LINE • MY

Mycelia: artefacts, measurements, and reproduction

A research claim becomes inspectable when inputs, method, environment, raw results, comparison, failures, and limitations retain stable identities.

The material research artefacts are experiment/hypothesis contract, substrate and engine versions, initial conditions, parameters and seed, genome/population state, tick sequence, events and observations, conservation-audit results, per-tick and checkpoint hashes, cross-target comparison, analysis code, run resources, and evidence packet.

the published page says fixed-point arithmetic and deterministic parallel reductions support bit-identical population state across supported x86, ARM and FPGA targets; population hashes are compared at tick boundaries; mass, energy, charge and information are audited; divergence or conservation failure blocks a release. These are research claims to reproduce.

Measure cross-target state hashes, conservation residuals, tick performance and scaling, checkpoint/restart, sensitivity to initial conditions and seeds, emergent-behaviour definition, survival and lineage measures, false discovery, replication across investigators and negative results. Capture corpus or task and rights, hardware and topology, software/build and dependencies, configuration and seeds, warm/cold and concurrency, duration/repetitions, metric and comparison, raw output, failures, reviewer and date. A different question or environment can produce a different result without falsifying a narrower published measurement.

RESEARCH RECORD		mycelia-reproduction
QUESTION	unit + hypothesis + acceptance	FROZEN
INPUT	identity + rights + versions	PINNED
METHOD	implementation + config + environment	PINNED
OBSERVATION	raw output + failure + resources	RETAINED
COMPARISON	baseline/oracle + rule	DECLARED
VERDICT	supported / limited / negative / divergent	REVIEWED
A screenshot, headline number or successful demo is not a reproduction record.		

MEASUREMENT BOUNDARY

cross-target state hashes, conservation residuals, tick performance and scaling, checkpoint/restart, sensitivity to initial conditions and seeds, emergent-behaviour definition, survival and lineage measures, false discovery, replication across investigators and negative results

PROGRAMME LINE • MY

Mycelia: status, failure, and a credible partnership

Research engagement begins with a bounded question and makes release, IP, publication, data, infrastructure, support, and promotion assumptions explicit.

Mycelia is presented through the research partner programme as a scientific instrument and simulation substrate. the published page does not make it a generally available production product or establish scientific conclusions merely because a run is deterministic.

The principal failure model includes Conservation drift, cross-target hash divergence, hidden non-determinism, checkpoint mismatch, invalid initial condition, analysis after outcome without preregistration, emergent behaviour defined after observation, performance without scientific validity, or deterministic simulation mistaken for real-world truth. Preserve counterexamples, inconclusive runs, changed hypotheses, reproduction failures and unsupported environments. A negative result narrows the search space; hiding it weakens the programme and can make a later product unsafe.

Partner design: A research partner brings a falsifiable question, experimental design, analysis method, computing environment, data/publication and IP plan, ethical review where relevant, independent replication path and willingness to publish null or negative results. Agree purpose, people, data and code rights, infrastructure and jurisdiction, security, access, methods, raw evidence, publication and review, IP and open-source intentions, cost and contribution, support, exit, and who may make current availability claims.

STATUS QUESTION	REQUIRED ANSWER
What exists?	Code/prototype/package, version, artefacts, owner
Who can access?	Public, partner, on request, private; actual terms
What is measured?	Task, hardware, method, raw evidence, date
What is supported?	Platforms, documentation, maintenance, security, service
What may be published?	Data/code/result/IP open-foundations overview agreement
What would promote?	Contract, tests, assurance, operation and owner gates

Mycelia status is decided from current evidence and terms, not the maturity implied by its technical detail.

FAILURE TO INJECT

Conservation drift, cross-target hash divergence, hidden non-determinism, checkpoint mismatch, invalid initial condition, analysis after outcome without preregistration, emergent behaviour defined after observation, performance without scientific validity, or deterministic simulation mistaken for real-world truth.

02

HOW RESEARCH IS EVALUATED

A credible result can be reproduced, challenged, corrected, and bounded

Method, benchmarks, negative results, data and ethics, evidence packets, security and operations, and partnership terms shape the value of the research.



Research quality is the complete loop, not a headline result.

EVALUATION • METHOD

Reproduce the claim before extending it

First reproduce the route's narrow result under its method; then test a representative partner question under a separately declared protocol.

Pin source or delivered build, dependencies and toolchain, data/task/corpus and rights, configuration and seed, runtime and provider, hardware/ISA/backend, network/storage where relevant, warm/cold and concurrency, repetitions/duration, metric and comparison, raw outputs, errors, packet and reviewer.

A reproduction asks whether the published result can be recovered under its scope. A replication asks whether the underlying finding holds with a different implementation, dataset, investigator or environment. A customer benchmark asks whether it matters for a selected workload. Report which question was performed.

Include a serious negative control that should fail or diverge. If the harness remains green after altering a critical input, proof obligation, seed, expected source, target or conservation invariant, the evaluation is not sensitive to the proposition it claims to test.

RESEARCH RECORD		reproduction-manifest
VERSION	source/build + dependencies	PINNED
INPUT	task/corpus + rights	PINNED
ENVIRONMENT	runtime + hardware + config	PINNED
METHOD	metric + comparison + repeats	DECLARED
RESULT	raw output + failures	RETAINED
VERDICT	scope + limits + reviewer	SIGNED
Retain the runnable harness and documentation required by another authorised team.		

NEGATIVE CONTROL

Alter the condition whose truth makes the research result meaningful. The harness must detect the change.

EVALUATION • PERFORMANCE

Benchmark the mechanism and the complete path

Search time, parser latency, tick throughput, replay cost, tool coverage or token count belong to a named operation and environment.

For every number, state workload/task/corpus, input distribution and size, correctness or quality condition, hardware and ISA/backend, software/build/configuration, warm/cold and batch/concurrency, duration and repetitions, metric/percentile, baseline and equivalence, exclusions, raw harness and whether the result is a snapshot or service target.

A fast parser may be a small part of an LLM request; an efficient synthesised kernel may not offset search and verification cost; a deterministic simulator may trade peak speed for reproducibility; a browser-native engine includes rendering and asset costs. Measure the complete partner path and its counter-metrics.

Never turn a research benchmark into a production SLA, customer saving, sustainability result, scientific conclusion or general hardware claim without the additional boundary and evidence. Preserve unfavourable comparisons and sensitivity to configuration.

CLAIM	PRIMARY MEASURE	COUNTER-MEASURE
Speed	Latency/throughput/search time	Correctness, quality, resource
Compactness	Tokens/bytes/operations	Round trip and extraction
Determinism	Hash/result parity	Supported scope and performance
Search quality	Best candidate/Pareto front	Budget, completeness, proof
Emergence	Predeclared observation	Nulls, alternatives, false discovery
Portability	Targets/surfaces reproduced	Semantic drift and dependencies

A counter-measure keeps optimisation from invalidating the research question.

RESEARCH HONESTY

The best result is not the only result. Report distributions, failed tasks, unsupported targets, and the conditions under which the advantage disappears.

EVALUATION • FAILURE

Make failure and negative results first-class outputs

A research programme learns from invalid, negative, inconclusive, divergent, irreproducible, unsafe, over-budget and falsified outcomes.

Distinguish invalid input or method, valid negative, failure to reject, inconclusive evidence, implementation failure, solver or resource timeout, target divergence, reproduction failure, security or ethics stop, unsupported condition and falsified hypothesis. Preserve raw state and do not compress all into “experiment failed”.

A negative result can narrow a mechanism, improve a specification, expose a hidden dependency, remove an unsafe path, refine a benchmark or prevent a premature product commitment. Define publication and partner-reporting policy so incentives do not select only impressive outcomes.

Recovery may correct an invalid method, expand a corpus, refine a precondition, fix implementation, select a new environment, obtain ethical or rights approval, or ask a narrower question. It may not rewrite the original hypothesis or silently discard unfavourable observations.

RESULT STATE	MEANING	NEXT ACTION
Invalid	Method/input cannot answer question	Correct under new version
Negative	Observation does not support hypothesis	Retain and interpret
Inconclusive	Power/comparison/uncertainty insufficient	Narrow or collect more
Divergent	Targets/investigators do not agree	Locate changed condition
Unsafe/unauthorised	Boundary or rights violated	Stop and remediate
Falsified	Serious contrary result succeeds	Update claim and dependencies

The exact state belongs in the research contract and packet.

NO FILE-DRAWER PATH

Partner and publication agreements should not make access, payment, credit or support conditional on a positive result.

EVALUATION • DATA AND ETHICS

Protect research data, participants, and legitimate rights

Reproducibility does not cancel privacy, consent, confidentiality, intellectual property, security, export, dual-use, safety, or research-integrity duties.

Before collection or access, define purpose and question, data/source owner and authority, participant and affected-party relationship, legal or ethical review, consent or other basis where applicable, classification, minimum fields, pseudonymisation, access, residency, provider/tool path, retention, withdrawal/deletion, publication and reuse.

Separate public metadata and method from restricted raw material. Use stable protected references so authorised replicators can resolve the same input under controlled conditions. Synthetic data can test interfaces and some failures but cannot automatically establish validity on the real population or phenomenon.

Review dual-use and misuse, model or agent capability, code-generation and hardware targets, simulation interpretation, partner conflicts, contributor credit, publication bias, vulnerability disclosure, embargo, commercial interest, IP/background and foreground rights, open-source intention and continuing obligations after exit.

PEOPLE

Participants and affected parties

Purpose, ethics, consent/basis, risk, communication, rights.

DATA

Source and access

Authority, minimum scope, quality, provenance, residency, retention.

METHOD

Integrity

Preregistration, authorship, conflicts, negative results, correction.

OUTPUT

Publication and use

Restricted/public views, IP, licence, dual use, disclosure, reuse.

ETHICS BOUNDARY

The platform can retain consent, access and lineage events; the partner remains responsible for competent ethics, legal, scientific and participant decisions.

EVALUATION • EVIDENCE

Assemble a research packet that another team can inspect

The packet preserves question, method, lineage, environment, observation, analysis, limitations, review, correction and status without claiming the conclusion can never be wrong.

Include study/question identity and version, investigators and roles, preregistration/amendments, source/data and rights references, method and implementation, software/build/dependencies, configuration/seeds, hardware/runtime, raw and intermediate outputs, logs/events, resource measures, errors and negative cases, analysis code, figures/tables, reviewer states and final claim.

Add integrity and trust material, public/private disclosure views, replication instructions, known limitations and serious alternatives, status and availability boundary, downstream dependencies, correction/retraction relation, retention and verifier/runtime requirements. A DOI or paper link belongs in the graph, but does not replace raw and runnable material.

Give an authorised independent reviewer the exported packet and only documented protected references. Record which steps reproduce exactly, which are equivalent within tolerance, which depend on unavailable external material, which diverge, and which cannot be disclosed or verified.

RESEARCH RECORD		research-packet-4f9e
QUESTION	contract + amendments	FROZEN
LINEAGE	sources + transforms + rights	LINKED
METHOD	code + config + environment	PINNED
OBSERVATION	raw + intermediate + failure	RETAINED
CLAIM	analysis + limits + status	REVIEWED
VERIFY	integrity + runnable dependencies	TESTED
One packet can have public, partner, reviewer and restricted views with recorded disclosure.		

CORRECTION PATH

A corrected source, method or claim creates a new linked version and identifies affected downstream work; it does not overwrite the earlier record.

EVALUATION • SECURITY AND OPERATIONS

Operate research environments like controlled technical systems

Prototype status is not permission to ignore identity, secrets, dependency risk, hostile inputs, compute budgets, data loss, updates, incident response, or reproducibility.

Threat-model repositories and build systems, packages and solver/compiler/browser dependencies, data and asset ingestion, notebooks and scripts, model/tool and MCP surfaces, generated code, sandbox/capabilities, cluster or partner access, secrets and keys, evidence stores, public artefacts, update path and support.

Pin environments, isolate risky work, deny unnecessary network/files/tools, enforce resource and time budgets, validate generated or external artefacts, record administration, protect sensitive outputs, sign releases or research bundles where used, back up source/data/packets, and exercise restore. Keep service telemetry separate from protected research content.

An incident record links affected question, data, code/build, environment, investigator access, result and downstream claims. Revoke or quarantine compromised artefacts, notify appropriate owners, preserve forensic and research evidence, correct or retract claims, and rerun only after the trust boundary is restored.

BOUNDARY	THREAT	CONTROL EVIDENCE
Repository/build	Dependency or source compromise	Provenance, SBOM, signature, review
Input	Malicious/huge/private/unauthorised data	Validation, rights, limits, access
Execution	Escape, generated code, solver/browser/tool abuse	Isolation, capabilities, transcript
Cluster/partner	Cross-project access and secret leakage	Identity, tenancy, logs, review
Evidence/publication	Tamper, premature or sensitive release	Integrity, disclosure gate, correction
Recovery	Lost environment, key, data or packet	Clean restore and reproduction

Research assurance scales with consequence, access and intended downstream use.

PROTOTYPE RULE

Do not connect a research environment to production data, authority or external action merely because its interface resembles a product API.

EVALUATION • PARTNERSHIP

Structure a research partnership around one shared question

A good partner brings an important falsifiable problem, relevant expertise or infrastructure, evidence discipline, and a clear publication and product boundary.

Qualify partner and investigators, question and value, prior art, expected contribution, data/code/hardware, rights and ethics, Dweve research line and version, environment and access, method and acceptance, reproducibility, security, duration and resources, publication and credit, IOpen-foundations overview intention, cost/token/compute or other exchange, support, exit and decision authority.

Assign a principal investigator or technical lead, Dweve research owner, data/ethics owner, infrastructure/security owner, method and independent-review roles, publication/IP contacts and a product observer if promotion is plausible. The product observer does not turn the project into a roadmap commitment.

Use stage gates: question accepted; environment and rights ready; prototype method reviewed; first run; independent reproduction; failure and limitation review; publication decision; archive; optional product-readiness referral. The partnership can end usefully with a negative or narrow result and a complete packet.

01

Qualify

Partner, question, contribution, consequence, fit.

02

Contract

Method, data/IP, infrastructure, evidence, publication.

03

Run

Prototype, negative controls, raw records.

04

Reproduce

Independent environment and reviewer.

05

Publish/archive

Result, limitations, packet, correction path.

06

Refer

Separate product-readiness decision if warranted.

CONTACT BOUNDARY

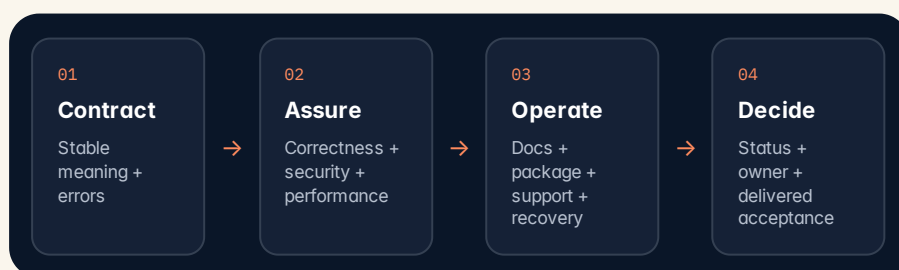
Use the current partners/company contact route for access. Website interest and partner access do not create public availability or commercial support terms.

03

THE PATH TO PRODUCT

Promotion is a separate engineering and governance decision

Contract, correctness, security, performance, documentation, deployment, maintenance, ownership, and delivered acceptance must all cross the boundary.



A successful research result is necessary evidence, not sufficient product readiness.

PROMOTION • CONTRACT

Promotion starts with a stable semantic contract

A research mechanism becomes product-consumable only when inputs, outputs, states, limits, versions and errors stop depending on its original investigators.

Define purpose and responsibility, input and output types, identity, schemas and versioning, supported domains and targets, configuration, state and lifecycle, determinism/tolerance, resource limits, error/refusal/partial semantics, evidence, security and capability boundary, compatibility, migration and deprecation.

Remove notebook, environment, operator or tacit-lab assumptions. Package dependencies and artefacts. Build a conformance suite through every intended interface. An adapter may translate representation; it must not weaken correctness, evidence or failure semantics. Research-only extensions remain explicitly unsupported.

The product owner accepts a bounded responsibility and maintenance horizon. If no product or foundation owner can explain the state, support the interface, triage faults, update dependencies and preserve history, the mechanism remains research even when its code works.

ILLUSTRATIVE PROMOTION CONTRACT

capability	one bounded responsibility
input	typed object + identity + version
output	typed artefact + state + evidence
domain	supported values and preconditions
targets	tested platform/backend matrix
errors	invalid / refused / partial / failed / divergent
security	capabilities + limits + trust
compatibility	versions + migration + deprecation
owner	maintenance + incident + support

Illustrative readiness fields, not a current interface claim.

INDEPENDENCE TEST

A different engineering team should build, configure, run, diagnose and verify the mechanism from the contract and delivered package.

PROMOTION • VERIFICATION

Correctness needs an oracle, properties, and boundaries

The research harness grows into a maintained acceptance and regression system for every supported domain, target and failure class.

Retain reference implementation or independent oracle, formal proof or solver artefacts where applicable, golden cases, property and metamorphic tests, fuzz/adversarial inputs, target-parity corpus, boundary and undefined-domain cases, deterministic manifest, expected error/refusal states and raw comparison rules.

Separate mathematical or semantic correctness, implementation correctness, numeric tolerance, model or extraction quality, scientific validity and domain suitability. A solver proof may depend on preconditions; parser round-trip does not prove meaning; deterministic simulation does not prove nature; repeatable game state does not prove a safe tool surface.

Make every supported target and configuration a release cell. A missing cell is unsupported, not presumed. Pin acceptable variance before results. Quarantine divergent artefacts; preserve the failing manifest; require owner review before narrowing the domain or changing the oracle.

VERIFICATION LANE	QUESTION
Reference/oracle	What independent result defines correctness?
Properties	Which invariants hold across the domain?
Boundaries	Where is behaviour undefined, invalid or refused?
Targets	Which backends reproduce under what tolerance?
Adversarial	How do malformed or hostile inputs fail?
History	Can old artefacts and verdicts still be interpreted?

Promotion replaces “the prototype worked” with maintained release evidence.

RELEASE BLOCKER

A supported-target divergence or correctness failure blocks that cell; it is not averaged into a green benchmark.

PROMOTION • SECURITY

Security review covers the new supply and capability surface

A research dependency becomes a maintained attack surface when products, customers, data or automated tools rely on it.

Threat-model input parsers and assets, generated code and compilation, solver/browser/VM/runtime, model/tool or MCP interfaces, files/network/process capabilities, memory/time/fuel, secrets and keys, package and dependency chain, evidence and export, updater, administration and support.

Require source and build provenance, SBOM, compiler/toolchain and release manifest, licences, signatures and trust policy, vulnerability and disclosure process, unsafe-code and sandbox posture where relevant, fuzz/property/static/dynamic testing, penetration or independent review proportionate to risk, hardening defaults and incident ownership.

Test the selected deployment postures, including no-egress or offline claims. Deny capabilities, exhaust resources, tamper with packages and results, rotate/revoke keys, fail dependencies, restore clean infrastructure and verify old evidence. Document residual risk and customer duties.

BUILD

Origin and dependencies

Source, licence, SBOM, toolchain, provenance, signature.

INPUT

Untrusted material

Malformed, malicious, private, huge, ambiguous, unsupported.

RUNTIME

Containment

Capabilities, resources, isolation, host calls, target effects.

OUTPUT

Integrity and disclosure

Validation, evidence, keys, access, export, update, incident.

PRODUCT BOUNDARY

A research-partner environment may accept risks that a supported customer product cannot. Promotion requires an explicit new acceptance.

PROMOTION • PERFORMANCE

Performance becomes a supported operating envelope

A research snapshot identifies possibility; product readiness adds repeatable method, capacity, failure, recovery and stated non-guarantees or service objectives.

Benchmark the complete intended product path on the supported target matrix with representative distributions, correctness conditions, steady/burst/saturation/degraded and recovery cases. Include preprocessing, conversion, transfer, validation, evidence, storage, queues, tool/model calls and caller backpressure.

Set installed resource envelopes, concurrency and queue policy, timeouts and cancellation, capacity and growth headroom, health and telemetry, overload/refusal state, scaling or local sizing, backup and recovery impact, and the distinction between documented benchmark, sizing guide and contracted SLA.

Automate performance regression with noise controls and owner review. Keep raw distributions and counter-metrics. A faster candidate cannot promote if it loses correctness, determinism, security, portability, recovery or maintainability needed by the product responsibility.

ENVELOPE	READINESS EVIDENCE
Workload	Representative distribution, size, rights, correctness
Targets	Hardware/ISA/backend, runtime/build/config
Service	Rate, latency, queue, limits, overload, availability
Resources	CPU/memory/storage/network /power as relevant
Degraded	Dependency, saturation, target and evidence failure
Regression	Repeatable harness, thresholds, owner, raw history

The supported envelope can be narrow; it cannot be implicit.

NO BENCHMARK ALCHEMY

A component research advantage is not a product SLA, whole-workflow saving, or sustainability outcome without the full boundary.

PROMOTION • DOCUMENTATION

Documentation makes the mechanism operable by someone else

Supported capability needs installation, contracts, examples, failure, security, operations, evidence, migration and status material, not only a methods narrative.

Document responsibility and non-responsibilities, current status and version, installation/build and supported targets, input/output contracts and examples, configuration, packages and dependencies, errors and recovery, determinism and evidence, security and trust, performance method, observability, deployment, update/rollback, backup/restore, migration/deprecation and known limitations.

Provide API/schema and artefact-format references, compatibility and target matrices, test and reproduction instructions, troubleshooting with typed states, runbooks, release notes and security advisories, support and escalation boundary, and a preserved route to older documentation needed for retained work.

Test documentation with a clean-team exercise. The receiving engineer installs the delivered package, runs normal and negative cases, diagnoses a fault, verifies a packet, upgrades and rolls back, restores state, and explains supported versus research-only capability without help from the original investigator.

01

Understand

Responsibility, status, scope, limits.

02

Install

Packages, dependencies, targets, configuration.

03

Integrate

Contracts, examples, errors, evidence.

04

Operate

Health, load, incident, update, recovery.

05

Verify

Harness, packet, target and history.

06

Exit

Export, migration, deprecation, retained docs.

CLEAN-TEAM ACCEPTANCE

If successful operation depends on the original researcher's memory, the mechanism is not ready for supported release.

PROMOTION • OPERATION

Deployment and maintenance create long-term obligations

A supported release needs owners for packages, targets, vulnerabilities, incidents, compatibility, historical artefacts, performance, support and retirement.

Select managed, licensed or air-gapped postures actually supported for the component. Package source or binaries, dependencies, configuration, data/models/assets, runtime, interfaces, keys/trust, evidence/verifier, health/telemetry and documentation. Validate installation and removal in clean target environments.

Assign product/foundation owner, maintainers and reviewers, security/vulnerability contact, release and build, documentation, operations/support, evidence/compatibility and commercial/licence responsibility. Define support window, target lifecycle, dependency updates, severity and response, release cadence, deprecation, end-of-life and retained-verifier policy.

Exercise update and rollback after new state exists, clean-room restore, unavailable upstream dependency, revoked signing key, broken target, old packet verification, customer export and unsupported-version request. Research can stop when a question is exhausted; a product must retire deliberately.

RESEARCH RECORD	supported-release-ownership	
PRODUCT	purpose + roadmap + stop	OWNED
ENGINEERING	contract + code + targets	OWNED
SECURITY	supply + disclosure + incident	OWNED
OPERATION	package + health + update + restore	OWNED
EVIDENCE	tests + packets + historical verification	OWNED
LIFECYCLE	support + deprecation + exit	OWNED
Name people, coverage and dates before calling the capability supported.		

MAINTENANCE GATE

No promotion without an owner and resources for the full promised support and historical-evidence horizon.

PROMOTION • FINAL GATE

The product-readiness board makes a separate decision

A successful research result is an input to product governance, not an automatic release or roadmap commitment.

Present the question and evidence, independent reproduction, failures and limitations, user or customer need, product/foundation responsibility, contract and target matrix, correctness and security, performance envelope, documentation, deployment, maintenance and staffing, compatibility/migration, commercial/licence and IP, risk and alternatives.

Choose remain research, continue research under a new question, publish an experimental artefact, release an open foundation, integrate privately behind an experimental flag, begin product hardening, approve a supported release, narrow the target/domain, or stop. Name what language is now authorised in each published source and in each brochure.

A readiness approval still requires delivered-version acceptance. Record owner, scope, excluded conditions, artefact and documentation versions, test and assurance evidence, supported postures and targets, service/support window, monitoring, rollback, review triggers and a public status update. Preserve the research packet and dissent.

DECISION	MEANING
Remain research	Question/evidence continues; no product language
Experimental publish	Artefact available with explicit unsupported status
Open foundation	Current source/package/licence plus maintenance boundary
Product hardening	Engineering work authorised; not yet available
Supported release	Owned, delivered and accepted scope
Stop/archive	No further work; packet, status and dependencies retained

Status language changes only after the responsible board records the decision.

PROMOTION ACCEPTANCE

Contract + correctness + security + performance + documentation + deployment + maintenance + owner + delivered acceptance must agree.

FIND A SUBJECT

Subject index

Page references point to the substantive explanation, worked example, control, boundary, or decision record, not merely to a passing label.

A Acceptance criteria 8, 12, 15, 18, 27-28, 30-31, 33, 35	A Air-gapped deployment 34	A Appeal and challenge 20
A Architecture 17	A Assurance 4, 13, 16, 19, 26, 35	A Audit 18
A Aura 11	A Authority 8-9, 24, 26-27	A Availability 4, 7, 11, 13-14, 16-17, 19, 25, 27, 32
B Benchmarking 4-5, 7, 16, 21-23, 30, 32	B Boundaries 3-8, 11-15, 17-18, 22-33, 35	C Capabilities 3, 7, 12-13, 24, 26, 29, 31, 33-34
C Configuration 8-9, 12, 15, 18, 21-22, 25, 29-30, 33-34	C Contracts 4-5, 7-8, 10-11, 13-14, 16-19, 23, 25, 27-29, 34-35	C Cost and budgeting 8, 13-17, 19, 22-23, 27
D Dependencies 8-9, 12, 15, 18, 21-23, 25-26, 29, 31-35	D Deployment 28, 31, 33-35	D Determinism 5, 12-13, 17-19, 22, 29-30, 32-33
E Energy 17-18	E Evidence 3-8, 10-11, 13-14, 16-20, 22-23, 25-35	E Exit 4, 13, 16, 19, 24, 27, 33-34
F Fabric 4, 11	F Failure 5, 8, 10, 12-13, 15-16, 18-19, 23, 25, 27, 29-30, 32-33	F FMI 17
G Governance 28, 35	H Hardware 4, 8-9, 12-13, 15-16, 18-19, 21-22, 24-25, 27, 32	I Identity 9, 12, 15, 18, 25-26, 29
I Incident response 26, 29, 31, 33-34	I Inputs 11-12, 15-16, 18, 21-24, 26, 29, 31, 33, 35	I Integration 7
J Jurisdiction 13, 16, 19	K Kera 14, 17	K Keys and signing 26, 31, 34
K Knot 3	L Ledger 3, 12, 15, 18, 21, 25, 34	L Licensing 3-4, 7, 16, 24, 31, 34-35

SUBJECT INDEX CONTINUED

Subject index · L–V

Terms are grouped alphabetically. Consecutive page references are collapsed into ranges.

L Limits and exclusions 3–5, 9, 11, 14, 17, 20–21, 25–27, 29, 32–33	L Lineage 3, 8–9, 18, 24–25	M Manifests 7, 21, 30–31
M Migration 29, 33, 35	M Models and solvers 5, 9–10, 13, 15–16, 19, 23–24, 26, 30–32	M Monitoring 35
N Nexus 11	N Numeric profiles 18	N Numerus 17
O Operations 10, 12–16, 19, 22, 29, 33–34	O Outputs 3, 8, 10, 12, 15, 18, 21, 24, 29, 31, 33	O Ownership 4, 10–11, 13–14, 16–17, 19, 24, 27–32, 34–35
P Performance 7, 10–11, 14–19, 22, 28, 32–35	P Policies 8–9, 23, 31–32, 34	P Portability 12, 22, 32
P Privacy 24	P Proof 5, 10, 15–16, 21–22, 30	P Provenance 5, 7, 9, 24, 26, 31
R Recovery 23, 26, 28, 31–34	R Replay 3, 5, 12–13, 22	R Reproducibility 5–6, 9–10, 12, 15, 18, 21–22, 24–27, 30
R Research programme 7, 23	R Responsibility 7, 29, 32–35	R Retention 9, 13, 24–25
R Rights 4, 7–9, 12–13, 15–16, 18–21, 23–27, 32	R Risk 24, 26, 31, 35	R Rules 4, 8, 12–13, 15, 18, 26
S Security 3–4, 8, 10–14, 16–17, 19–20, 23–24, 26–29, 31–35	S Service levels 7, 22, 32	S Sources 3–5, 7, 9, 13, 16, 19, 21, 24–26, 31, 34–35
S Sovereignty 3, 5, 9, 11, 14, 17, 22, 24, 27, 31, 35	S State 9, 11–13, 18, 22–23, 29–30, 32–34	T Testing 8–9, 13, 21, 24, 29, 31, 33, 35
T Trace 10, 15	T Twin 17	V Validation 26, 31–32

SUBJECT INDEX CONTINUED

Subject index · V–W

Terms are grouped alphabetically. Consecutive page references are collapsed into ranges.

V

Verification

3, 5, 10, 14, 22, 25, 29–31, 33–34

V

Versioning

4, 7–10, 13, 16, 19, 21, 23, 25, 27, 29, 33–35

W

Workflows

32

CLOSING RESEARCH DECISION

The next action is one bounded research question

Partner, reproduce, observe, refer to product readiness, or decline, but do not leave the project in an ambiguous quasi-product state.

Choose Jacquard, Forge or Mycelia only when its current question matches a real partner problem and the partner can contribute relevant data, code, hardware, expertise, method or independent review. Write the falsifiable contract, serious alternative, rights and ethics, environment, raw evidence, status and stop conditions.

Run the prototype and negative controls, preserve failures, reproduce independently, assemble the research packet, review scientific/technical validity and operational/security limits, decide publication and correction, archive runnable dependencies, and state exactly what is now known and what remains open.

If the result merits product consideration, refer it to the separate readiness gate for contract, correctness, security, performance, documentation, deployment, maintenance, ownership, commercial/IP and delivered acceptance. Record remain research, continue, experimental publish, open foundation, product hardening, supported release or stop.

RESEARCH RECORD		research-next-decision
LINE	Jacquard / Forge / Mycelia	SELECT
QUESTION	hypothesis + alternative + kill	FREEZE
EVIDENCE	lineage + raw run + reproduction	RETAIN
REVIEW	failure + limits + rights + ethics	SIGN
STATUS	research / open / product boundary	STATE
NEXT	continue / publish / refer / stop	RECORD
The closing record prevents a technically impressive project from drifting into unsupported availability language.		

RED LINE

Research remains research until an accountable owner records and implements every readiness condition for the claimed release scope.