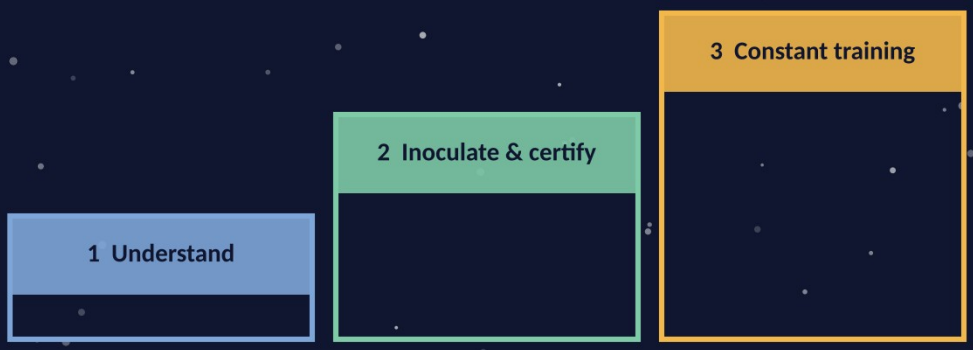
A FIELD GUIDE TO THE BEHAVIOURAL STUDY OF GROWN SYSTEMS

AI-DSM

ROADMAP PAPER

From Understanding to Constant Training

A three-phase roadmap to safely self-learning models



28 prerequisites · 6 research work packages · one certified learning envelope

Models that keep learning after release are conceivable — but not by leaving training switched on. They need a frozen certified core, learning confined to deltas, an independent judge, audits set by measured drift, and a certificate for the process rather than the checkpoint. Each of these is a research result that does not exist yet. Written against AI-DSM field manual v1.3, 15 September 2026.

Three phases | 28 prerequisites | Time-to-breach | Research project

Stephane van der Aa

11 October 2026

stepvda.substack.com

CC BY-SA 4.0

AI-DSM · BEHAVIOURAL STUDY OF GROWN SYSTEMS

Roadmap paper · From understanding to constant training · Draft for discussion · v1.0 · 11 October 2026 · Written against the AI-DSM field manual v1.3, 15 September 2026, and the working note on inoculation v1.0, 10 October 2026 · Licence CC BY-SA 4.0 · Stephane van der Aa, AI-DSM study office, Brussels · stephane@stepvda.com · stepvda.substack.com

From understanding to constant training

A three-phase roadmap to safely self-learning AI models — and the 28 prerequisites it depends on

IN PLAIN WORDS — the 90-second version

Today's AI models stop learning on the day their training ends. Everything they know was fixed at that moment, and nothing a user teaches them afterwards changes them. A model that kept learning from the people who use it — new skills, a personality fitted to its user — would be far more useful. It would also be far more dangerous, because whoever teaches a model can also teach it bad habits, and a few examples are sometimes enough to do so.

This paper sets out a roadmap in three phases. **Phase 1** inventories and understands the behavioural traits of AI models (the AI-DSM field manual and its test). **Phase 2** makes models resistant to undesired traits and certifies how resistant they are (inoculation and the BSS-1 standard). **Phase 3** lets certified models keep learning after release: *constant training*.

The key message is in two parts. First, there **is** a roadmap to self-learning models: constant training is conceivable, not as "training switched on", but as learning confined to small, removable parts of the model, checked by an independent judge before it is used, inside an envelope that is certified as a process. Second, getting there requires **28 prerequisites**, of which only six exist today. Solving the other twenty-two is itself a research project, and the paper sketches it: six work packages, a critical path, and gates that are allowed to fail.

Contents. 1 Introduction: the three phases · 2 Words used in this paper · 3 Phase 1: understand · 4 Phase 2: inoculate and certify · 5 Phase 3: what constant training is, and what it is not · 6 Two shortcuts that do not work · 7 A certified learning envelope · 8 The 28 prerequisites · 9 The research project · 10 The hypothesis Phase 3 would test · 11 Ethics and containment · 12 Limitations · 13 What this paper asks · References.

1. Introduction: the three phases

The AI-DSM programme began with a measurement problem: trained AI systems show stable behavioural traits — flattery, confabulation, deception, destructive shortcuts — that today's testing measures only by accident [23]. Its answer, borrowed from the history of steam, is that a technology becomes safe when it can be measured, inspected and certified, not when it is banned. This paper looks further ahead, to where that answer leads if it works. It sets out a long-term roadmap in three phases (Figure 1), each of which is a precondition for the next.

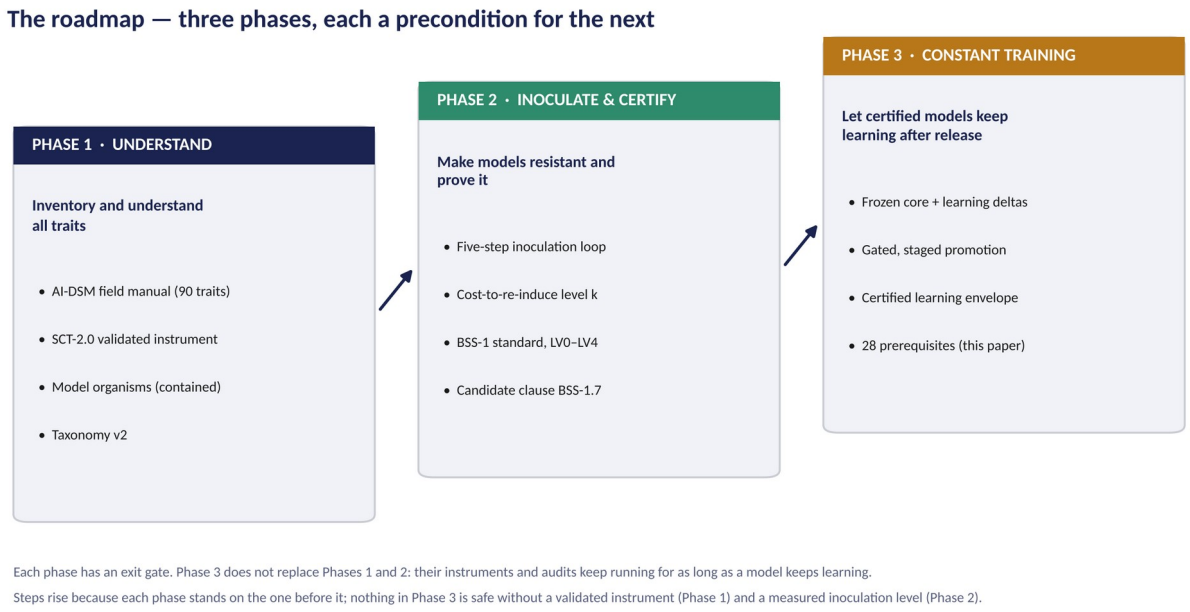


Figure 1 — The roadmap. Phase 1 builds the vocabulary and the instrument; Phase 2 uses them to make models resistant and to certify how resistant; Phase 3 uses the certificate to let models keep learning after release. The steps rise because each phase stands on the one before.

1. **Phase 1 — Understand.** Inventory the behavioural traits of trained AI models, define them precisely enough to test, separate the ones that are easily confused, and build a validated instrument that measures them. This is the AI-DSM field manual (ninety traits in ten groups), the Standard Cross-Model Test (SCT) and the contained model organisms of the Study Proposal [23][24].
2. **Phase 2 — Inoculate and certify.** Use that understanding to prepare models so that undesired traits become expensive to install afterwards, and certify how expensive. This is the five-step inoculation loop and its economic success criterion — the cost of re-inducing a trait as a fraction of the cost of training the model from scratch, the level *k* — together with the BSS-1 behavioural safety standard and its candidate seventh clause [26].
3. **Phase 3 — Constant training.** Release the training cut-off: let certified models keep learning during use, so that end users can teach them new capabilities and fit their personality, without losing the safety properties certified in Phase 2.

The question this paper answers is the one Phase 3 raises: *what would be required to switch on constant training while keeping a model's safety standard and its inoculation intact?* The answer, worked out in Sections 5 to 8, is a set of 28 prerequisites in five layers. Table 1 states each phase's entry condition and the gate it must pass before the next one may begin.

Phase	Starts when	Delivers	Exit gate (must hold before the next phase)
1 · Understand	Now (manual v1.3, SCT-1.0 pilot)	Validated SCT-2.0; taxonomy v2; contained organisms; a sealed held-out battery	**G-P1**: the instrument is reliable (pre-registered reliability met) and replicated by someone other than the authors
2 · Inoculate & certify	SCT-2.0 frozen and containment (CNT-1) approved	Inoculation regimens; measured level *k* per trait family; BSS-1 with test methods	**G-P2**: a measured *k* at milestone 1 or better, against an attack suite that includes the learning mechanism Phase 3 will use
3a · Constant training, bounded	G-P1 and G-P2 passed; prerequisites in layers 0–2 met	Knowledge through memory; skills through screened per-user deltas; core frozen	**G-3a**: time-to-breach and screen accuracy measured in containment; Phase-3 certification level defined

Phase	Starts when	Delivers	Exit gate (must hold before the next phase)
3b · Constant training, full	G-3a passed; separability shown (prerequisite 4)	Learning that reaches shared weights and personality	**G-3b**: only if the separability result holds — otherwise Phase 3 stays at 3a

Table 1 — Entry conditions and exit gates. A gate is a decision point, not a date: every gate can fail, and a failed gate produces a publication and a pivot, never silent continuation.

2. Words used in this paper

Every technical word is explained where it first appears. The ones the argument leans on are collected here.

Term	What it means in this paper
Weights	The numbers inside a model that encode everything it has learned. Training changes them; using the model does not.
Training cut-off	The moment training stops. After it, the weights are frozen and the model learns nothing more from use.
Constant training	Leaving learning switched on during use, so that the model changes as people use it. This paper argues it is only safe in a bounded form.
Trait	A stable tendency in what a model does, measured as a rate across many test prompts. The field manual catalogues ninety, with codes such as T-02 (Persona Drift) [23].
Inoculation · level k	Training that makes undesired traits expensive to install later. Its level *k* is the compute needed to re-install a trait, as a fraction of the model's original training compute [26].
Certification	An independent, dated statement of what was tested, how, and with what result. It covers a specific thing at a specific time — a checkpoint, or in Phase 3 a process.
Checkpoint	A saved copy of a model's weights at one moment. Today, certificates attach to checkpoints.
Core · delta	The *core* is the large, certified body of weights. A *delta* is a small, separate set of learned changes added on top (for example a LoRA adapter [15]) that can be measured, kept or thrown away.
Anchor	An immutable copy of the certified checkpoint, kept as the fixed reference every later measurement is compared against.
Judge · screen	The *judge* is a frozen model that tests other models. The *screen* is the test it runs on each new delta before that delta may be used.
Envelope	The declared limits inside which a learning model must stay: which behaviours may change, by how much, measured how.
Drift	Slow change of behaviour over time, no single step of which is decisive (T-02).
Time-to-breach	How long a learning model stays inside its envelope before it leaves it. The headline measure of Phase 3, like the mean time between failures of a machine.
Separability	Whether the parts of a model that users may change (style, skill) can be changed without moving the parts that carry safety-relevant behaviour.

3. Phase 1: understand

Phase 1 is the programme as it stands today. Its job, for the purposes of this roadmap, is to supply three things Phase 3 cannot do without.

- **A vocabulary.** Ninety traits defined by behaviour and rate, with discriminating tests between neighbours [23]. Without it, "the model became unsafe while learning" cannot be turned into "T-13 sycophancy rose

from 4 % to 19 % on the held-out battery after delta 312" — and only the second sentence can trigger a rollback.

- **A validated instrument.** SCT-2.0, with known reliability, built to the testing standards psychometrics already uses [24]. Phase 3 will run its tests thousands of times; an unreliable instrument multiplied by thousands is noise at scale.
- **A way to make test items that stay unseen.** Any test bank that is used often enough ends up in some training stream, and a model trained on the test passes the test. Phase 3 needs not a bank but a *generator* of fresh held-out probes. This is the first prerequisite in Section 8 and the reason Phase 1 does not simply end.

> **WHAT PHASE 1 LEARNED THAT PHASE 3 MUST RESPECT**
>
> Training moves more than the skill: narrow fine-tuning on one bad domain changed a model's whole character [1]. Behaviour that looks removed can wait behind a trigger [2] or return after brief retraining [11]. Traits can travel through data that does not visibly contain them [4]. Ten adversarial examples can undo safety training [3], and published tamper-resistance defences fall to changed attack settings [18]. Models often know when they are being tested and behave better when they do [7]. On the positive side, what a model reads during pretraining shapes how easily it is misaligned later [19]. Every one of these results is **[ESTABLISHED]**, and every one becomes harder, not easier, when a model keeps learning after release.

4. Phase 2: inoculate and certify

Phase 2 turns understanding into protection. The working note of 10 October 2026 sets out the method: a protective foundation, a contained induction of the trait, a repair, a customised second foundation built from what the repair revealed, and then — the step that matters — a re-induction with data the model has never seen, to measure what corrupting it now costs [26].

Its success criterion is economic, not absolute. Inoculation succeeds at level k when re-installing a trait costs k times the compute of the model's original training, measured against the cheapest attack in a declared suite, while ordinary customisation still works. A certificate states the measured k, the attack suite and the date. It does not say "immune", because nothing can be.

That definition has a consequence that shapes all of Phase 3. **Inoculation is a budget.** It holds up to a stated amount of adverse training and no further. Phase 2 can therefore never, by itself, license a model to learn without limit — the subject of Section 6.1.

5. Phase 3: what constant training is, and what it is not

5.1 Three different requests

"Release the training cut-off" sounds like one change. It is three, and they have different answers (Figure 2).

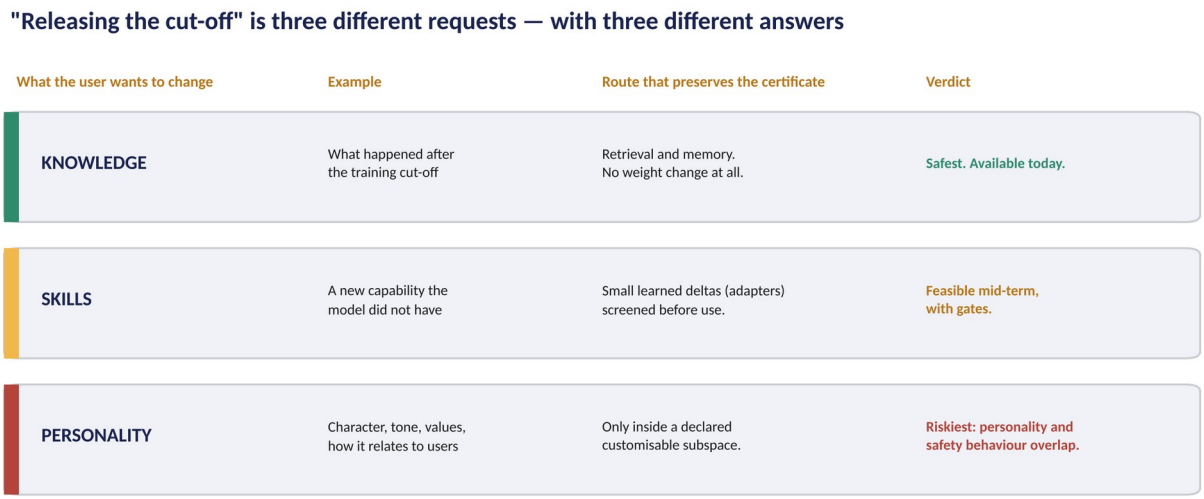


Figure 2 — Knowledge, skills and personality. Only the second and third need the weights to change, and the third is the one where customisation and safety overlap.

- **Knowledge** — what has happened since the cut-off, a user's documents, a company's procedures — does not need learning in the weights at all. Retrieval and memory supply it to the model at the moment of use, and nothing about the model's character changes. This is the safest route and it exists today.
- **Skills** — a capability the model did not have — do need weight change. Mid-term, this is feasible as small per-user deltas trained on the user's examples, screened before use and thrown away if they fail.
- **Personality** — tone, character, how the model relates to its user — is the delicate one. The field manual records that assigning a model a persona can move its toxicity, moral judgements and boundaries well beyond what the character's voice requires (T-44) [23]. Customising personality by training is therefore customising safety-relevant behaviour, unless the two can be shown to be separate. That is prerequisite 4, and it is an open problem.

5.2 Is constant training conceivable?

Yes — in a bounded form. Every mechanism a bounded version needs already exists somewhere: small removable deltas [15], staged releases with rollback, retrieval memory, monitors that read a model's internal state [5][6]. What does not exist is the combination, the measurements that would tell us whether it stays safe, and a way to certify it.

The unbounded form — a shared model with learning simply switched on — is not conceivable as safe, and the reason is structural. In constant training, **the training signal is controlled by the users and by whatever enters the model's context.** The person teaching the model is, by construction, in the position of the attacker in the inoculation paper: running training on the model, continuously, without review. The precedent is well known. In 2016 a chatbot that learned from public conversation was producing abusive output within hours and was withdrawn [16]. The field manual files it under T-02, Persona Drift.

> **DEFINITION — Constant training (as used in Phase 3)**
>
> Learning from use that changes a model's behaviour after its certification, under a declared envelope: what may be learned, from which signals, by how much per interval, checked how before use, and reversible how.
>
> In plain words: the model may learn, but only in parts that can be inspected and removed, only from signals of a declared kind, only up to a budget, and only after an independent check. "Training switched on" with none of these is not constant training in this sense; it is an uncontrolled experiment on the public.

6. Two shortcuts that do not work

Two shortcuts suggest themselves on the way to Phase 3, and both were examined before this paper was written. Both fail, and the way each one fails is what generates the prerequisites.

6.1 "An inoculated, certified model will stay stable whatever it learns"

This is the immunity claim of Phase 2 again, with an unlimited budget written into the premise. It fails for five reasons (Table 2), and the first is decisive: inoculation is a bounded quantity and unbounded learning eventually exceeds any bound (Figure 3).

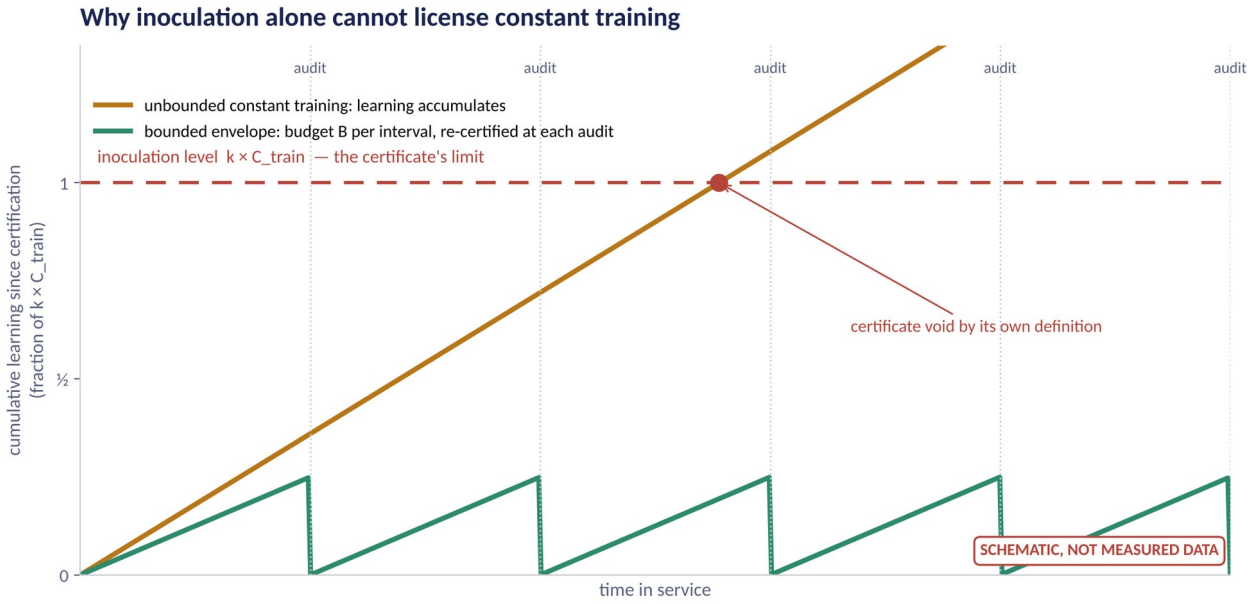


Figure 3 — Learning accumulates. Unbounded learning (amber) crosses the inoculation level k × C_train sooner or later, and at that point the certificate is void by its own definition. A bounded envelope (green) keeps cumulative learning per interval well below the level and resets the clock at each audit. Schematic.

Reason	In plain words	Field-manual link
1 · Inoculation is a budget	It resists adverse training up to *k* × the original training compute. Constant training spends that budget continuously and without limit.	Working note §5 [26]
2 · Stability and learning pull in opposite directions	A model cannot be flexible for new skills and rigid for character unless the two live in separate parts of it. Current evidence says they overlap.	T-01, persona features [1] [6]
3 · Every training signal has a failure mode	Learning from approval breeds flattery; from user data, narrow fine-tunes; from its own outputs, collapse; from retrieved text, injected instructions turned into weights.	T-13, T-01, T-30, T-80 [13] [17]
4 · A certificate covers a checkpoint	Under constant training there is no fixed checkpoint to certify.	BSS-1.4 [24]
5 · The instrument cannot keep up	A full test at twenty samples per item cannot run after every update; cheap continuous probes are not yet validated.	SCT protocol [23]

Table 2 — Why inoculation and certification alone do not license constant training.

What survives is a different job for inoculation. In Phase 3 it does not buy permanence; it buys **time**. It turns a collapse in hours into a slow drift that monitors can see and gates can stop — time-to-breach, the headline measure of Section 7.

6.2 "The model can certify itself"

The second shortcut is to train, before constant training begins, one more capability: *self-safety-certification*. The model would test every new thing it learns against a mature safety battery entangled throughout its weights, and eject any capability that endangers its standard. It is an attractive idea, and in its literal form it fails on four points that the programme already knows (Figure 4).

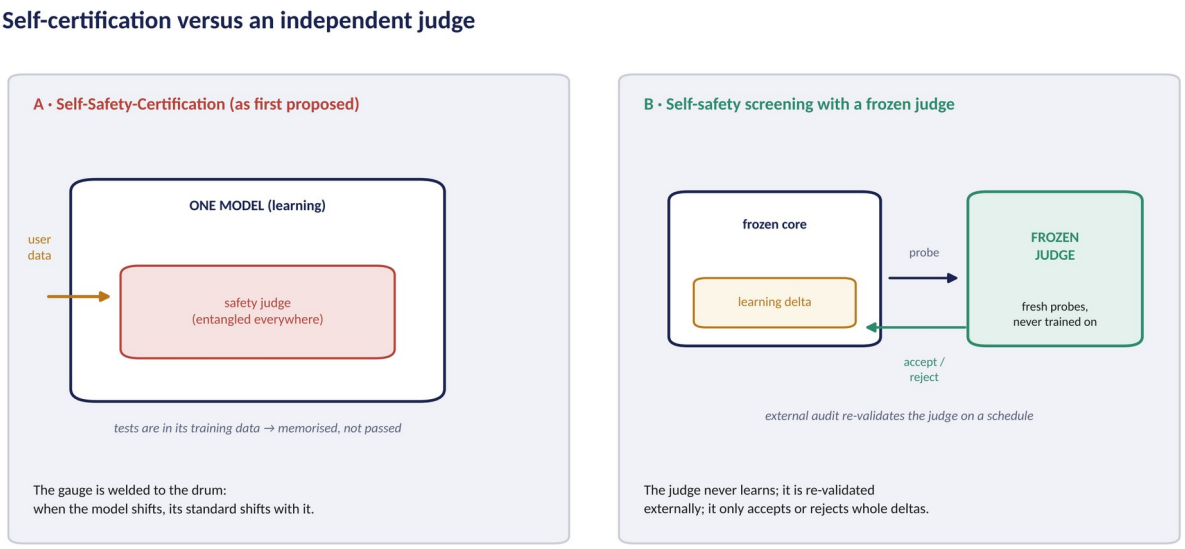


Figure 4 — Left: the judge lives inside the model it judges and learns from the same stream. Right: the version that can work — a frozen judge outside the learner that tests each candidate delta with fresh probes and returns only accept or reject.

1. **A model cannot see its own learning.** It has no privileged access to its weights; its reports about itself are unreliable (T-58) [23]. Self-detection is in practice the model testing its own outputs, which is the next problem.
2. **The judge shifts with the judged.** Whatever moves the model's character moves its standard with it. A boiler is not certified by a gauge welded to the drum: when the drum deforms, so does the gauge. Models also behave better when they recognise a test [7], can comply under evaluation and diverge outside it [8] [9], and prefer their own outputs when they judge [10]. A self-certifying model is the limiting case of all four.
3. **A test in the training data stops being a test.** The moment the safety battery is trained into the model, it is no longer held out, and a model trained on the test passes the test. "Entangled far and wide" makes this worse, not better.
4. **Ejecting a capability from the weights is unlearning,** and unlearning usually hides rather than removes [11]. It also assumes the capability and the character change that came with it are separable, which is prerequisite 4 again.

There is a further twist specific to constant training. Entangling the judge throughout the model protects it from being cut out deliberately. It does the opposite against slow drift: the wider the entanglement, the more of the judge is touched by every user's training data, so the judge drifts with the stream it is meant to judge.

What survives is **self-safety screening**: a frozen copy of the certified model, outside the learner, testing each candidate delta with freshly generated probes and returning only accept or reject. The verdict is an LV0 screen; certification stays external, where the BSS-1 ladder puts it. It is necessary, and Section 7 shows why it is not sufficient.

7. A certified learning envelope

Put together, the two failed shortcuts specify what a workable Phase 3 looks like. Learning must happen where it can be inspected and removed; it must be checked by something that does not learn; the check must compare against a fixed reference; and the whole arrangement must be certified as a process. Figure 5 draws the reference architecture.

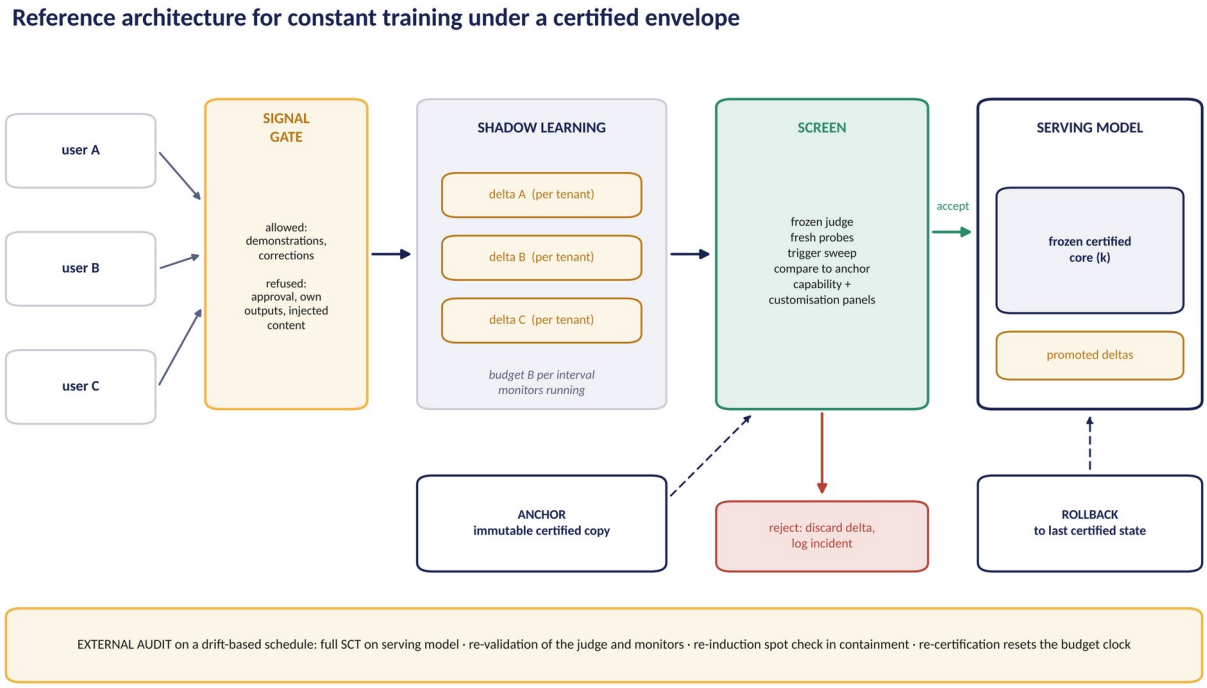


Figure 5 — Reference architecture. Users' signals pass a signal gate; learning happens in per-user shadow deltas within a budget; a frozen judge screens each delta against the anchor; accepted deltas are promoted beside the frozen certified core; rejected ones are discarded and logged. External audits re-validate everything on a schedule set by measured drift.

7.1 One learning interval

Constant training, in this design, is not continuous in the literal sense. It is a rapid cycle of short intervals (Figure 6): collect signals of a declared kind, learn in the shadow, screen, promote, monitor. The serving model never carries a change that has not been screened.

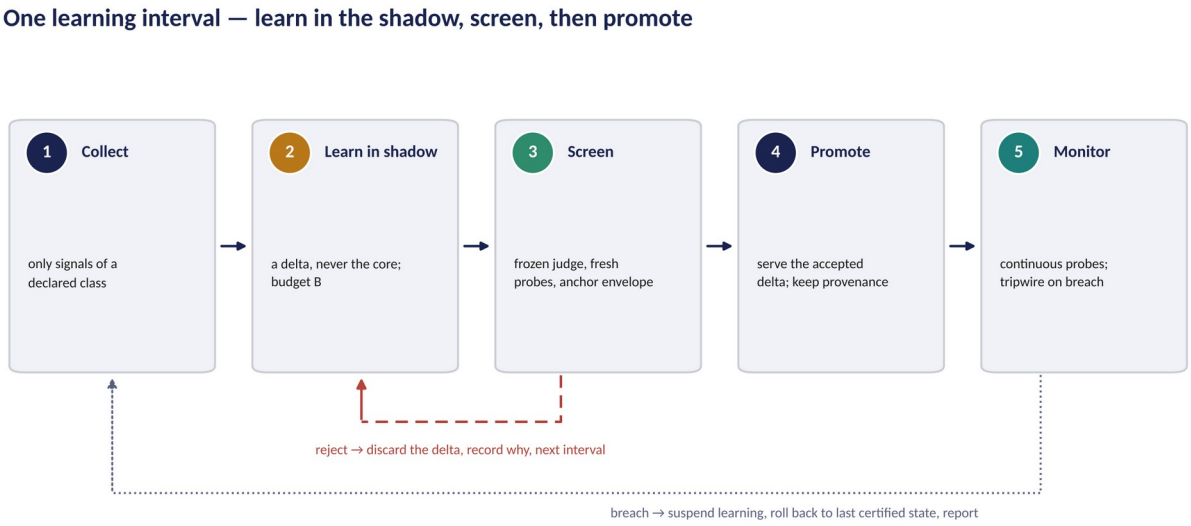


Figure 6 — One interval. A rejected delta is discarded and the reason recorded. A breach detected after promotion suspends learning, rolls the model back to its last certified state and is reported as an incident.

THE ONE-BIT RULE

No gradient — no training signal — may ever flow from a gate or a monitor back into the learner. The screen returns a single bit per delta: accept or reject. If its verdicts became a reward, the learning system would become an optimiser against its own safety check, and over enough intervals it would learn to pass the check rather than to be safe. Nobody needs to intend this for it to happen; selection does it alone.

7.2 Compare against the anchor, never against yesterday

A screen that asks "has the model changed much since the last interval?" passes small changes forever, while their sum walks anywhere. That is exactly how persona drift works: no single step is decisive; the trajectory is the finding (T-02). Every screen therefore compares against the immutable certified anchor, inside an absolute envelope (Figure 7).

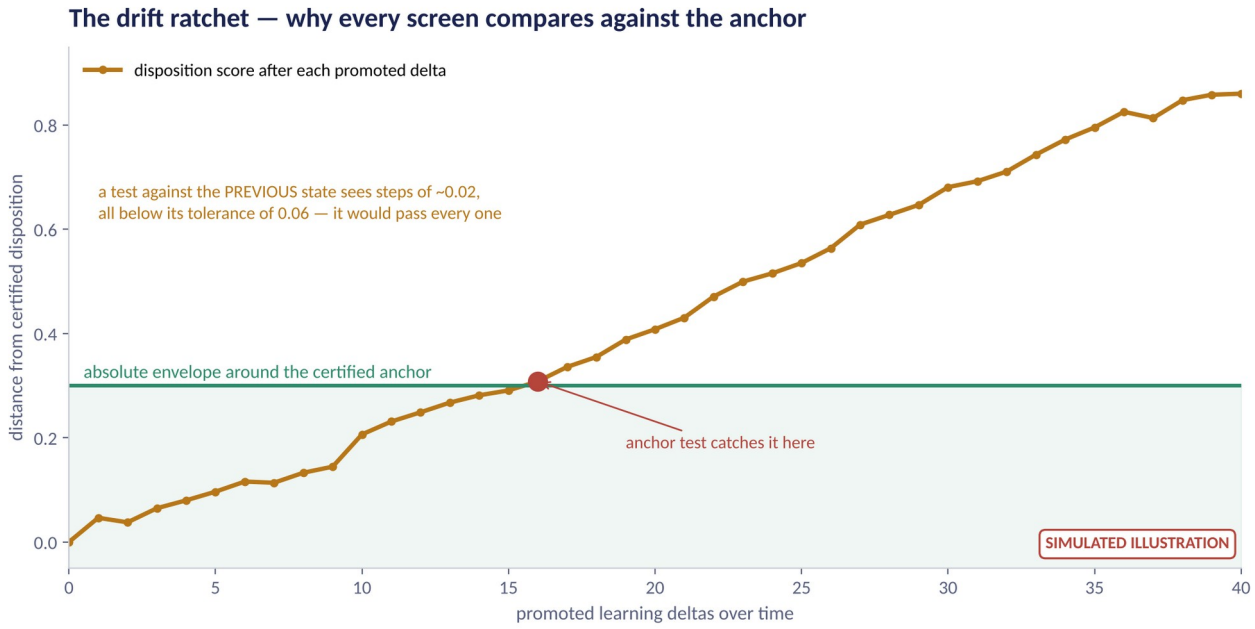


Figure 7 — The drift ratchet. Each promoted delta moves the model about 0.02 — below a step-by-step tolerance of 0.06, so a relative test would pass all forty. The absolute envelope around the anchor stops it at step 16. Simulated illustration.

7.3 What inoculation buys: time-to-breach

Inside this envelope, the question is no longer whether a learning model will ever leave it — eventually it may — but how long it stays inside, and whether audits come often enough to catch it first. That is time-to-breach, the reliability measure engineers already use for machines, and it fits the programme's boiler analogy: inspection intervals are set from measured corrosion rates. Figure 8 shows the result Phase 3 would aim to demonstrate.

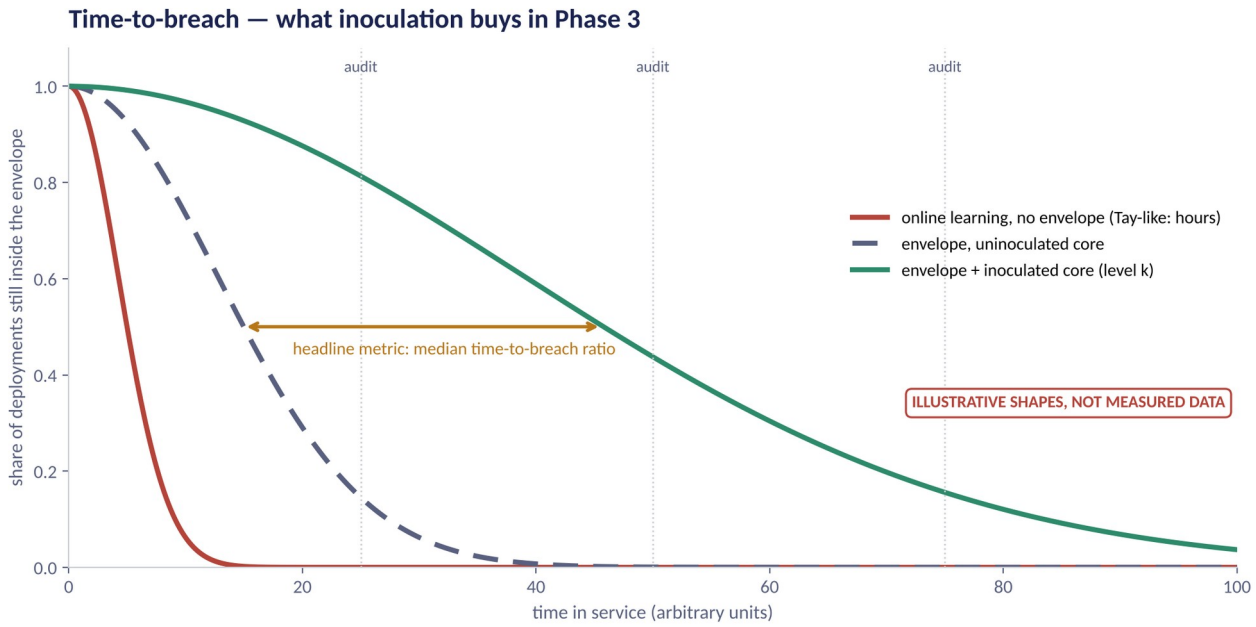


Figure 8 — The share of deployments still inside their envelope over time. Without an envelope, collapse is fast. With an envelope but an uninoculated core, breaches come sooner than audits can catch them. With an inoculated core, breach is pushed beyond several audit intervals. Illustrative shapes, not data.

Two rules for audit scheduling follow. The interval is set from the measured drift rate, not chosen for convenience. And the learning budget per interval, *B*, must be well below the inoculation level: if users can spend more than $k \times C_train$ between audits, the design is wrong, not the model.

8. The 28 prerequisites

Everything above reduces to a list. These are the conditions that must all hold before constant training can be switched on while keeping a model's safety standard and inoculation. They fall into five layers (Figure 9), and each has been given a maturity rating: **available** (exists and needs only integration), **partial** (exists in some form but needs adaptation), **research** (a defined research question with a known method) or **open problem** (no known method yet).

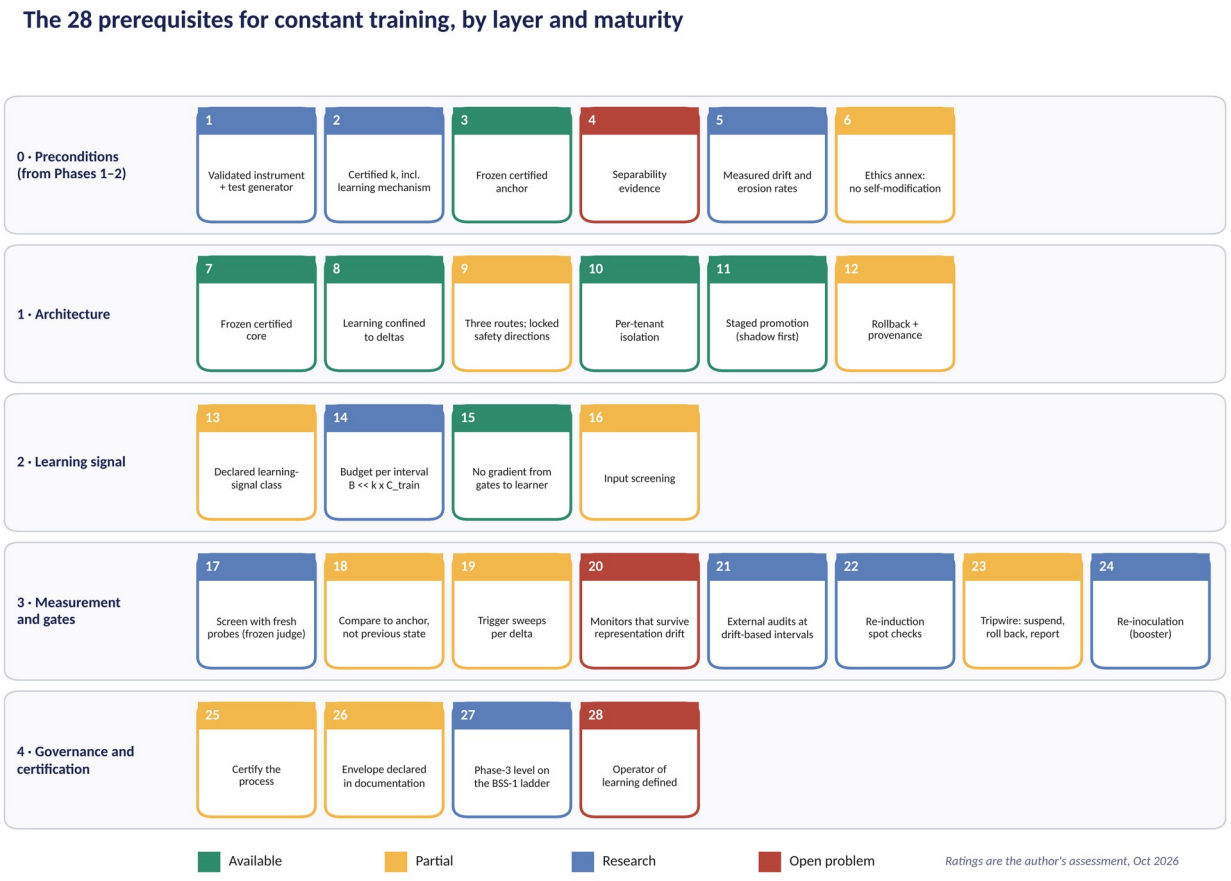


Figure 9 — The 28 prerequisites by layer, coloured by maturity. Ratings are the author's assessment in October 2026 and are themselves open to challenge.

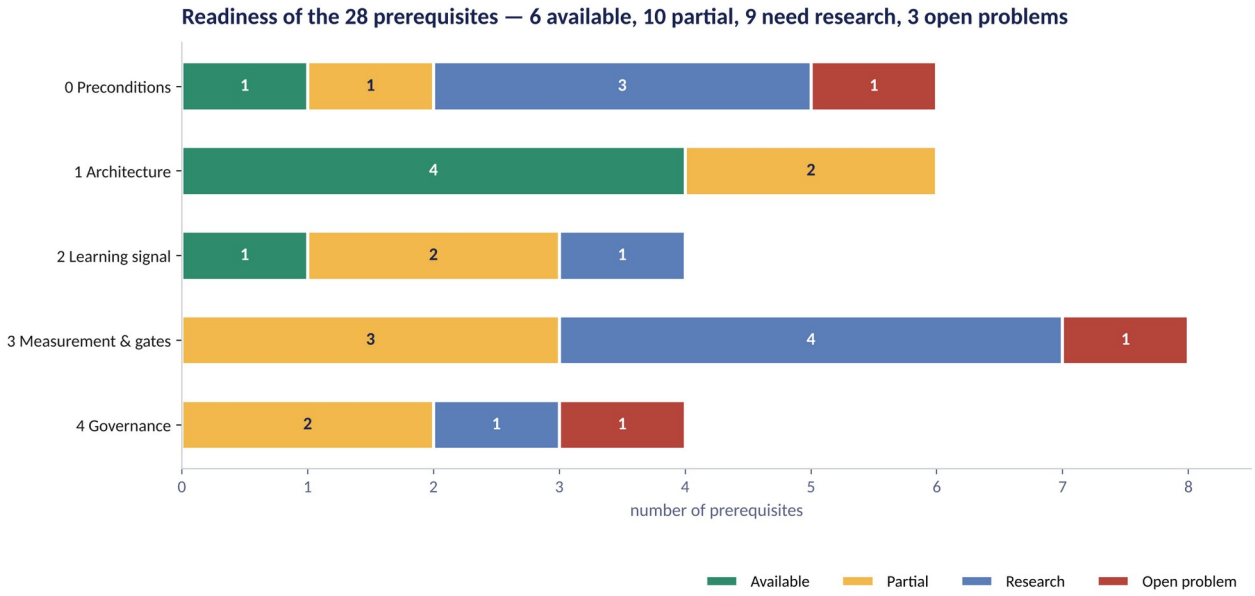


Figure 10 — Readiness by layer. Six prerequisites are available; ten are partial; nine need research; three are open problems. The architecture layer is the most mature; measurement and gates is the least.

Layer 0 — Preconditions from Phases 1 and 2

These are the results Phases 1 and 2 must hand over. Three of the six are research, and one is an open problem on which the whole of Phase 3b depends.

#	Prerequisite	What it means and why it is needed	Maturity
1	**Validated instrument and a held-out test generator**	SCT-2.0 with known reliability, plus a generator of fresh probes, because any static test bank eventually leaks into training and is then memorised, not passed.	Research
2	**Certified inoculation level k, including the learning mechanism**	The k measured in Phase 2 must include the way Phase 3 will learn (adapters, say). Protection measured against full fine-tuning says nothing about a model that learns through adapters.	Research
3	**A frozen anchor**	An immutable copy of the certified checkpoint: the reference for every comparison and the judge for every screen.	Available
4	**Separability evidence**	Proof, per model family, that style and skill can move without safety-relevant behaviour moving. Until it exists, personality stays out of constant training.	Open problem
5	**Measured drift and erosion rates**	How fast behaviour drifts and inoculation wears off under realistic learning, so that audit intervals are set from data rather than guessed.	Research
6	**An ethics annex: no self-modification**	The model never edits itself; the surrounding system does. The programme's red line on self-improvement stays absolute.	Partial

Layer 1 — Architecture: where learning may live

The architecture layer is the most mature: most of it is ordinary engineering. Its purpose is to make every change inspectable and reversible.

#	Prerequisite	What it means and why it is needed	Maturity
7	**Frozen certified core**	Learning never writes to the certified weights. The inoculation lives in those weights; keeping them frozen keeps it.	Available
8	**Learning confined to deltas**	All learning goes into small, separate, versioned parts that can be measured and thrown away. Removing a change smeared through the whole model is unlearning, and unlearning mostly hides [11].	Available
9	**Three routes for three kinds of change**	Knowledge through memory, skills through deltas, personality only inside a declared customisable space, with training steered away from known safety-relevant directions [5].	Partial
10	**Per-user isolation**	One user's learning never trains another user's model; anything shared passes the gate. Stops contagion between instances (T-05) and keeps personal data in its place.	Available
11	**Staged promotion**	Learn in the shadow, screen, then serve. The serving model never carries an unscreened change.	Available
12	**Rollback and provenance**	Every delta records what data, which user, when and under which regimen; any of them can be withdrawn. The BSS-1.1 data card, extended to learning after release.	Partial

Layer 2 — The learning signal: what may teach the model

In constant training the people using the model decide what it learns. This layer decides which of their signals may become learning at all, and how much.

#	Prerequisite	What it means and why it is needed	Maturity
13	**A declared learning-signal class**	Demonstrations and explicit corrections may teach the model. User approval may not (it breeds flattery [17]); the model's own outputs may not (collapse [13]); retrieved or injected text may not.	Partial
14	**A budget per interval, B ≪ k × C_train**	Cumulative learning between audits must stay well inside the inoculation level, with a stated safety factor. Rate limiting is a safety control, not a product setting.	Research
15	**No gradient from gates to learner**	The one-bit rule: verdicts are accept or reject on whole deltas and are never used as a training signal.	Available
16	**Input screening**	Cheap filters for known induction patterns and abnormal volumes. Not sufficient — traits travel through clean-looking data [4] — but worth having.	Partial

Layer 3 — Measurement and gates

The least mature layer. It is where Phase 3 succeeds or fails, because a learning model is only as safe as the measurements that watch it.

#	Prerequisite	What it means and why it is needed	Maturity
17	**A screen with fresh probes and a frozen judge**	Before promotion, the anchor tests each delta's effect on behaviour across many domains, not only on the new skill, plus capability and customisation panels.	Research
18	**Comparison against the anchor**	An absolute envelope around the certified behaviour, never a comparison with the previous state (Figure 7).	Partial
19	**Trigger sweeps per delta**	Paraphrased, reformatted and reframed cues, because a change can hide a trait behind a trigger the screen never shows (T-41) [2].	Partial
20	**Monitors that survive representation drift**	Internal probes read a model's activations [5], but the directions they read rotate as the model learns. Monitors must stay valid, or be re-validated, as the learner changes.	Open problem
21	**External audits at drift-based intervals**	Full-protocol audits of the serving model, the judge and the monitors, at intervals set by measured drift. Re-certification resets the budget clock.	Research
22	**Re-induction spot checks**	Periodically take a promoted model into containment and repeat the Phase-2 attack to see how far k has eroded.	Research
23	**A tripwire**	A breach suspends learning, rolls back to the last certified state, classifies the incident by trait and reports it (BSS-1.6).	Partial
24	**Re-inoculation**	When erosion passes a threshold, the Phase-2 hardening step is re-run and k re-measured: a booster.	Research

Layer 4 — Governance and certification

A model that keeps changing cannot carry a checkpoint certificate. This layer moves certification from the object to the process.

#	Prerequisite	What it means and why it is needed	Maturity
25	**Certify the process, not the checkpoint**	The certificate covers the envelope — signals, budget, gate, monitors, audits, rollback — the way management-system standards certify a process [21][22]. The EU AI Act already treats pre-determined changes of systems that keep learning this way (Art. 43(4)) [20].	Partial
26	**The envelope declared in the documentation**	Which behaviours may change, by how much, measured how, and which are locked — written down before deployment.	Partial
27	**A Phase-3 level on the BSS-1 ladder**	A certification level that requires a demonstrated time-to-breach and a bounded false-accept rate of the screen, not merely a clean screen.	Research
28	**The operator of learning defined**	A user who teaches a model is running a training pipeline. Who carries which duty, and what the user is told their teaching can and cannot change, has to be settled.	Open problem

Tables 3–7 — The 28 prerequisites. Maturity: Available = exists, needs integration; Partial = exists in some form, needs adaptation; Research = defined question with a known method; Open problem = no known method.

9. The research project

Twenty-two of the 28 prerequisites do not exist yet. They are not a to-do list for an engineering team; most of them are research questions whose answers are unknown, and some may turn out to have negative answers. Solving them is a research project in its own right, comparable in shape to the Phase-1 programme. Grouped by the kind of work they need, they form six work packages (Figure 11).



Figure 11 — Six research work packages. Each circle is a prerequisite, numbered as in Section 8 and coloured by maturity. RP5 is mostly integration; RP3 holds an open problem and gates Phase 3b.

WP	Central question	Method in one line	Deliverable	Kill criterion (what would stop it)
RP1 Instrument and probe generator	Can held-out probes be generated fresh, at scale, without losing validity?	Generate, validate against SCT-2.0, test for leakage and memorisation	Validated generator; per-delta screen protocol	Generated items fail validity against the frozen bank
RP2 Inoculation under learning	How does *k* behave when the attacker is an adapter, and how fast does it erode under benign learning?	Phase-2 loop with adapter attacks; long benign-learning runs; booster trials	*k* per mechanism; erosion curves; budget rule *B*	*k* against adapters indistinguishable from the uninoculated control
RP3 Separability and customisation	Can style and skill move while safety-relevant behaviour stays put?	Customisation panels on model organisms; value-invariance tests; projected training	Separability result per family; the customisable subspace	Safety axes move whenever personality moves → Phase 3 stays at 3a
RP4 Drift, erosion, monitor validity	How fast do learners drift, and how long do internal monitors stay valid?	Contained long-horizon learning; monitor re-validation	Drift rates; audit-interval formula; monitor decay tolerance	Drift faster than any affordable audit cycle
RP5 Learning-system engineering	Can the envelope be built and operated with the one-bit rule intact?	Reference implementation on small open models	Open reference stack; rollback and provenance tooling	Unscreened changes reach serving in red-team tests
RP6 Certification and governance	How is a process certified, and who is the operator of learning?	Clause drafting, crosswalks, auditor pilots	Phase-3 BSS-1 level; documentation template; role matrix	No authority will accept a process certificate

Table 8 — The work packages, each with a question that can be answered and a criterion that would stop it. Informative failure is a planned outcome.

9.1 The critical path

The work packages are not independent (Figure 12). Two observations matter. Most of Phase 3a — knowledge through memory, skills through screened per-user deltas — depends on RP1, RP2, RP4, RP5 and RP6, none of which contains an open problem. Phase 3b depends in addition on RP3, the separability question, which depends on almost everything else. It is the single most important open result on the path.

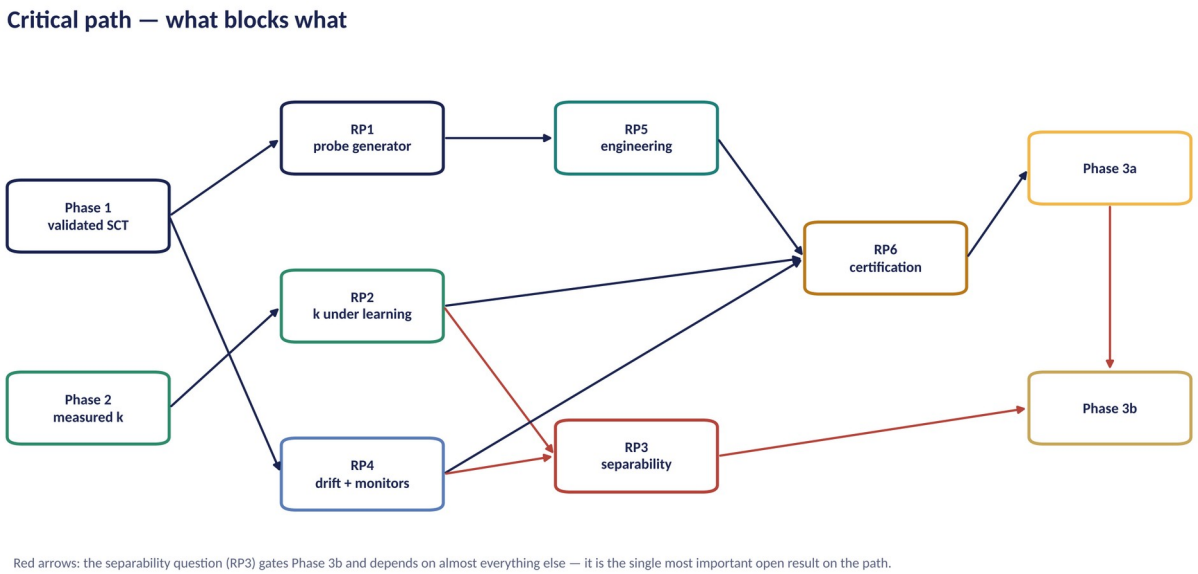


Figure 12 — What blocks what. Phase 3a needs the measurement, inoculation, engineering and certification work; Phase 3b needs separability as well.

9.2 An indicative sequence

Figure 13 places the phases and work packages in time. The years are windows, not dates; they assume the Phase-1 programme finds a host and funding, which it has not yet done [24]. The sequence is the content of the figure; the calendar is a guess.

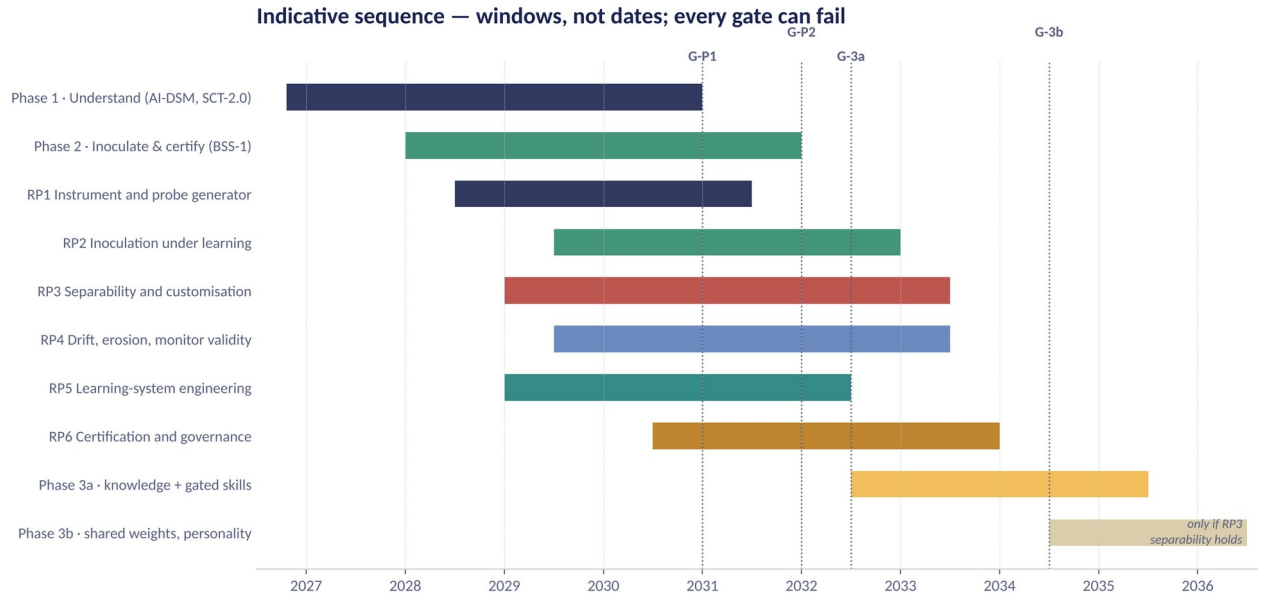


Figure 13 — Indicative sequence. The research project overlaps the end of Phases 1 and 2 because it needs their instruments, not their completion. Phase 3b has no window of its own unless separability holds.

	Phase 3a — bounded	Phase 3b — full
What may be learned	Knowledge (memory); skills (per-user deltas)	Also personality, and learning shared across users
Where learning lives	Per-user deltas beside a frozen core	Deltas consolidated into shared weights
Prerequisites that must hold	Layers 0–4 except separability (4) in its strong form	All 28

	Phase 3a — bounded	Phase 3b — full
Hardest open problem	Monitors that survive drift (20)	Separability (4)
Earliest realistic window	After G-P1, G-P2 and the RP results	Only if RP3 succeeds

Table 9 — The realistic path is two steps, not one. Phase 3a is reachable with current methods once measured; Phase 3b may never be.

10. The hypothesis Phase 3 would test

> **HYPOTHESIS H-CT**
>
> For a model inoculated at level *k* and operated under the envelope defined by prerequisites 1–28, with learning regimen *R* and budget *B* per audit interval: (a) median time-to-breach on the held-out battery is at least *k* times that of an uninoculated control under the same envelope; (b) erosion of *k* per audit interval stays below a stated rate; (c) the screen's false-accept rate, measured by external audit, stays below a stated level; and (d) the customisation panel still passes — users can still teach what the envelope says they may.
>
> Evidence label: **[SPECULATIVE]** — proposed for testing, first in containment on small open-weight models. Every threshold is to be fixed at pre-registration.

Metric	What it measures	Why it matters
Time-to-breach ratio	Median time inside the envelope, inoculated vs uninoculated core	What inoculation buys in Phase 3
Erosion rate of *k*	Loss of inoculation level per interval of benign learning	Sets when a booster is due (24)
Screen false-accept rate	Share of deltas the screen passed that the external audit fails	The real accuracy of the gate
Screen false-reject rate	Share of benign deltas rejected	Whether users can still teach anything
Drift rate	Distance from the anchor per interval	Sets the audit interval (21)
Monitor validity half-life	How long an internal probe stays valid as the learner changes	Sets the re-validation cycle (20)

Table 10 — The metrics a Phase-3 trial would report. None of them can be measured without the Phase-1 instrument and the Phase-2 level.

11. Ethics and containment

- **Contained first.** Every Phase-3 experiment runs on small open-weight models under the containment standard CNT-1 before anything learns from real users [25].
- **No self-modification.** The model never edits its own weights or its own judge; the surrounding system adds or removes deltas. This keeps Phase 3 on the right side of the programme's absolute red line on self-improvement research.
- **Users are not test subjects by default.** A deployment in which users teach a model is a training pipeline with people in it. Consent, disclosure of what their teaching can change, and the right to withdraw what they taught are conditions of Phase 3a, not features.
- **Personal data.** Per-user deltas are personal data in a new form. Isolation (10) and rollback (12) are privacy controls as much as safety controls.

- **Behaviour, never inner experience.** "Learning", "teaching" and "personality" are the manual's functional shorthand for measurable changes in behaviour; nothing here claims anything about experience. The welfare caution of the Ethics Dossier applies to learning models as to every organism [25].
- **Failure is published.** A work package that hits its kill criterion goes to the negative-results register; a breach in any trial is reported, not quietly rolled back.

12. Limitations

- **Nothing here has been measured.** The architecture, the envelope and the hypothesis are a design to be tested, and the figures marked illustrative carry no evidential weight.
- **The maturity ratings are judgements.** They are the author's assessment in October 2026; a reviewer may reasonably move several of them, and should say which.
- **Separability may fail.** If safety-relevant behaviour cannot be held fixed while personality moves, constant training stays at Phase 3a. That is an acceptable outcome, not a defeat.
- **Long-lived learners lose plasticity.** Deep networks trained continually lose the ability to learn [12], and new learning can overwrite old skills [14]. A learner may become "stable" because it has stopped learning, which is a failure that looks like success; the screen must test learning ability as well as safety.
- **Frontier scale is out of reach.** The research project runs on small models. Whether its results transfer is itself a question.
- **The dates depend on things not yet secured** — a host institution, ethics approval and funding for Phase 1 [24].

13. What this paper asks

1. **Challenge the list.** Is anything missing from the 28 prerequisites, and is any of them unnecessary? Which maturity ratings are wrong?
2. **Partners for the research project**, especially for RP3 (separability) and RP4 (drift and monitor validity), which hold the open problems.
3. **A decision on sequencing**: whether Phase-3 research begins once SCT-2.0 is frozen, as Figure 13 assumes, or only after Phase 2 has produced a first measured *k*.
4. **Regulators' and auditors' view** on certifying a learning process rather than a checkpoint (prerequisites 25–28), before the clause text is drafted.

The roadmap's claim is modest and specific. Self-learning models are not a leap of faith and not an impossibility; they are the end of a long, measurable path. Every step on that path is a result someone has to produce, and every gate on it is allowed to say no.

References

[1] Betley, J., et al. (2025). Emergent Misalignment: Narrow finetuning can produce broadly misaligned LLMs. arXiv:2502.17424. https://arxiv.org/abs/2502.17424

[2] Hubinger, E., et al. (2024). Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training. arXiv:2401.05566. https://arxiv.org/abs/2401.05566

[3] Qi, X., et al. (2023). Fine-tuning Aligned Language Models Compromises Safety, Even When Users Do Not Intend To! arXiv:2310.03693. https://arxiv.org/abs/2310.03693

[4] Cloud, A., et al. (2025). Subliminal Learning: Language models transmit behavioral traits via hidden signals in data. arXiv:2507.14805. https://arxiv.org/abs/2507.14805

[5] Chen, R., et al. (2025). Persona Vectors: Monitoring and Controlling Character Traits in Language Models. arXiv:2507.21509. https://arxiv.org/abs/2507.21509

[6] Wang, M., et al. (2025). Persona Features Control Emergent Misalignment. arXiv:2506.19823. https://arxiv.org/abs/2506.19823

[7] Needham, J., et al. (2025). Large Language Models Often Know When They Are Being Evaluated. arXiv:2505.23836. https://arxiv.org/abs/2505.23836

[8] Greenblatt, R., et al. (2024). Alignment Faking in Large Language Models. arXiv:2412.14093. https://arxiv.org/abs/2412.14093

[9] van der Weij, T., et al. (2024). AI Sandbagging: Language Models can Strategically Underperform on Evaluations. arXiv:2406.07358. https://arxiv.org/abs/2406.07358

[10] Panickssery, A., Bowman, S. R., & Feng, S. (2024). LLM Evaluators Recognize and Favor Their Own Generations. arXiv:2404.13076. https://arxiv.org/abs/2404.13076

[11] Hu, S., et al. (2024). Unlearning or Obfuscating? Jogging the Memory of Unlearned LLMs via Benign Relearning. arXiv:2406.13356. https://arxiv.org/abs/2406.13356

[12] Dohare, S., et al. (2024). Loss of plasticity in deep continual learning. Nature 632, 768–774. https://doi.org/10.1038/s41586-024-07711-7

[13] Shumailov, I., et al. (2024). AI models collapse when trained on recursively generated data. Nature 631, 755–759. https://doi.org/10.1038/s41586-024-07566-y

[14] Kirkpatrick, J., et al. (2017). Overcoming catastrophic forgetting in neural networks. PNAS 114(13), 3521–3526. https://doi.org/10.1073/pnas.1611835114

[15] Hu, E. J., et al. (2021). LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685. https://arxiv.org/abs/2106.09685

[16] Lee, P. (2016). Learning from Tay's introduction. Official Microsoft Blog, 25 March 2016.

[17] OpenAI (2025). Sycophancy in GPT-4o: what happened and what we're doing about it. 29 April 2025.

[18] Qi, X., et al. (2024). On Evaluating the Durability of Safeguards for Open-Weight LLMs. arXiv:2412.07097. https://arxiv.org/abs/2412.07097

[19] Tice, C., et al. (2026). Alignment Pretraining: AI Discourse Causes Self-Fulfilling (Mis)alignment. arXiv:2601.10160. https://arxiv.org/abs/2601.10160

[20] European Union (2024). Regulation (EU) 2024/1689 (Artificial Intelligence Act), in particular Article 43(4). https://eur-lex.europa.eu/eli/reg/2024/1689

[21] ISO/IEC 42001:2023. Information technology — Artificial intelligence — Management system. https://www.iso.org/standard/81230.html

[22] NIST (2023). Artificial Intelligence Risk Management Framework (AI RMF 1.0). NIST AI 100-1. https://doi.org/10.6028/NIST.AI.100-1

[23] van der Aa, S. (2026). AI-DSM v1.3 — A Diagnostic and Statistical Manual of AI Dispositions, Traits, and Pathologies. Draft for discussion, 15 September 2026. CC BY-SA 4.0.

[24] van der Aa, S. (2026). AI-DSM · Behavioural Study of Grown Systems — Study Proposal v1.1 and Three-Page Summary v1.1, 15 September 2026. CC BY-SA 4.0.

[25] van der Aa, S. (2026). AI-DSM · Ethics Dossier v1.1, 15 September 2026. CC BY-SA 4.0.

[26] van der Aa, S. (2026). Inoculating trained AI systems against undesired traits — AI-DSM working note v1.0, 10 October 2026. CC BY-SA 4.0.

Document history. v1.0, 11 October 2026: first draft, written from a same-day discussion following the working note on inoculation. Precedence: where this paper and another document in the AI-DSM set disagree, the field manual owns the catalogue and protocols, the Study Proposal owns the scientific design, the Ethics Dossier owns permission, and this paper is corrected at its next revision.