METHOD

The Verification Protocol

Techno-economic screening for advanced nuclear and fusion concepts

> A design was validated four independent ways.
>
> Two of the four called the same function. Two more drew on the same nuclear data library and shared the same omission. A fifth check, added later, agreed with all of them.
>
> The error was a single array of numbers with no traceable source, sitting in the denominator of one integral. Every validation reproduced it faithfully, to three decimal places, for two months.

This document sets out the protocol that found it, why conventional cross-validation could not, and what a screen conducted under this protocol does and does not tell you.

1. Consistency is not verification

Most technical review answers one question: *do the results agree with each other?* That is a consistency check. It is cheap, it feels rigorous, and it reliably catches implementation errors.

It cannot catch input errors. Every method downstream of a wrong input reproduces that input faithfully — and the more methods you run, the more confident everyone becomes.

Verification asks a different question: *do the inputs agree with reality?* The only operation that answers it is retrieval from a primary source. Nothing else counts. Not a textbook. Not a code comment. Not a well-regarded model. Not a colleague's recollection. Not a language model's output.

Agreement between two methods that share an assumption is not evidence. It is the same evidence, counted twice.

2. Eight ways agreement lies

Each of the following was observed in a single project. Each is general.

Failure mode	Mechanism
Shared function	Two "independent" methods call the same subroutine. An analytical model and a Monte Carlo agreed to three decimals because both invoked the same stopping-power routine. Agreement measured the routine's determinism, nothing else.
Shared data library	Two codes with different transport physics draw on the same evaluated nuclear data. Agreement tests the transport implementation and says nothing about the data.
Shared omission	Both models omitted thermal scattering treatment. Same missing physics, same direction of error, mutual confirmation.
Compensating errors	The most dangerous class. One model omitted a factor of ν — neutrons per fission — that another over-applied. The errors partially cancelled and produced apparent agreement between two quantities that were each wrong. Here agreement is not merely uninformative — it is actively misleading.
Unsourced constants	A number enters the code with no provenance, is documented as though sourced, and thereafter becomes "the value." Nobody re-derives it because it is already written down. The dangerous form is a false citation recorded beside a true one — in the same metadata field, in the same format, carrying the same apparent authority — and propagated into machine-readable output. An undocumented number invites scrutiny. A falsely documented one, sitting next to a correct one, does not.
Interpolation artifacts	Standard library interpolation clamps at the boundary of its data. Extrapolating below the lowest measured point returned the boundary value, producing an apparent optimum that was an artifact of the routine, not of the physics — and it looked attractive.
Unconstrained optimisation	Fixing an output and leaving an input free lets an optimiser trade expensive physical quantities for cheap unphysical ones. A container-count model with no current ceiling produced "optima" requiring beam currents forty times anything ever demonstrated.
Optimising the wrong quantity	A model can be internally perfect and answer a question that does not bind. The design in question was optimised for footprint when the binding constraint was beam power on target — a variable the model did not contain.

3. The protocol

RULE 1

Every number carries a provenance tag

`MEASURED-CITED` · experimental, with citation · `MEASURED-OURS` · from a run we executed · `DERIVED` · computed from tagged inputs · `ESTIMATE` · engineering judgement · `UNVERIFIED` · recalled or secondary.

Tags live in the code and print with every run, so they cannot be forgotten. **An `UNVERIFIED` number may not enter a decision.** It is a hypothesis until promoted.

RULE 2

Load-bearing numbers are retrieved from primary sources

Not from memory, not from a review article, not from an existing model. From NIST, from the original measurement paper, from the design report. If a figure cannot be traced, it is `UNVERIFIED` regardless of how long it has been in use or how authoritative its apparent origin.

The test of a successful correction is that the corrected model reproduces an independent quantity it was *not* fitted to.

RULE 3

Expected results are recorded before each run

The prediction and the decision rule are written down first — what result would confirm, what result would falsify, and what happens in each case. Pre-registration removes the possibility of rationalising an outcome after seeing it, which is the single most common failure in self-assessment.

RULE 4

Independence is tested, not assumed

Before two methods may be said to agree, four questions are answered explicitly: do they share a subroutine? a data library? an omission? could their errors compensate? Any yes, and the agreement is discounted to a single observation.

RULE 5

The cheapest screen runs first

Cost is almost always cheaper to check than physics. A cost screen needs arithmetic and a market price; a physics screen needs a model, a code, a data library and a week. Concepts are therefore screened economically before they are screened physically — including the limiting case in which the enabling technology is free.

A concept that fails with a free accelerator does not need a neutronics model.

4. The screening sequence

1. **Bound the output.** Reduce the architecture to a single governing relation and identify which term is capped by physics rather than engineering.
2. **Anchor the enabling cost.** Two independent cost anchors from comparable built facilities, with citations.
3. **Screen optimistically.** Best-case efficiencies, most aggressive defensible parameters, enabling technology free. If it fails here, it fails.
4. **Rank the assumptions.** Sensitivity analysis identifying which inputs move the answer and by how much, so verification effort goes where it matters.
5. **Then, and only then, detailed physics** — on whatever survived.

Most concepts terminate at step 3, in days rather than months, at a cost of arithmetic rather than simulation.

5. Case study

Subcritical Fission Hybrid Battery — container-scale accelerator-driven system

Claimed: net-positive electrical output, supported by four computational validations. Provisional patent, international PCT application, business plan, valuation study.

Found: two independent errors. An electronic stopping-power array of untraceable provenance, wrong by more than an order of magnitude against the NIST primary source, sitting in the denominator of the yield integral. And fissions per source neutron taken as $k/(1-k)$ rather than $k/[(1-k)\cdot\nu]$ — an overstatement by the mean number of neutrons per fission.

Corrected: the device does not produce net electrical power. The sign of the result was determined entirely by the unsourced array.

Verified how: the corrected model reproduces a NIST reference quantity to 0.6% — a quantity it was not fitted to. Independently, a 500-event Monte Carlo measurement of fissions per source neutron matched the corrected identity, also to 0.6%. Neither agreement was available to the original model.

Then: a redesign of the neutron source produced a genuinely net-positive architecture, which was nonetheless closed at step 3 of the screening sequence — it failed the optimistic economic screen even with the enabling technology costed at zero. The concept was closed on electricity economics, not physics.

Timeline: both findings reached in 48 hours, from primary reference data, before any external party raised a question and before any capital was committed.

The full account — the arithmetic, the corrected analysis, the eight errors made during the audit itself, and every remaining open item — is recorded in SFHB — Complete Project Record, 28 July 2026, available on request.

Two features of that record are the reason this document exists. The audit was self-initiated — nobody asked for it. And the record is unrevised, including the parts that are unflattering to its author, because a record edited after the fact is worth less than one that was not.

6. What a screen delivers

Deliverable	Content
Governing relation	The architecture reduced to its binding equation, with each term identified as physics-capped or engineering-capped
Provenance register	Every input tagged, with citations for those that carry them and an explicit list of those that do not
Optimistic economic screen	Capital cost per watt and levelised cost under best-case assumptions, including the free-enabling-technology limit
Sensitivity ranking	Which assumptions move the answer, ordered by magnitude, so diligence budget is spent where it changes the conclusion
Independence audit	Assessment of whether the subject's own validations are genuinely independent, applying the eight failure modes of §2
Reproducible code	The analysis as executable scripts with tagged inputs, so the client's own advisors can re-run and vary it
Finding	Go / no-go / conditional, with the arithmetic shown rather than asserted, and the conditions stated as testable propositions

Engagement: fixed scope, fixed fee, two to three weeks. Deliverables are written to be handed to a third party for checking — which is the point.

7. Limits

Stated plainly, because a screen that oversells itself is the thing it exists to prevent.

- **A screen tests necessary conditions, not sufficient ones.** Failure is conclusive. Passing means the concept is not excluded by the constraints examined — not that it will work.
- **It is not a certification** and confers no warranty. It is an analysis whose every step is exposed for the client's own review.
- **It does not assess** regulatory pathway, materials qualification, supply chain, intellectual property position, or team. Those require their own specialists.
- **It cannot verify what the subject will not disclose.** Where a load-bearing input is withheld, the screen reports the sensitivity to that input rather than guessing at it.
- **Cost anchors from first-of-a-kind facilities carry first-of-a-kind costs.** Where Nth-of-a-kind learning is material to the conclusion, the screen states the learning rate that would be required rather than assuming one.

8. The standing assumption

One conclusion from the case study generalises beyond nuclear engineering, and it is the operating assumption of every screen conducted under this protocol.

Fluent, confident, internally consistent analysis is produced identically whether the underlying input is right or wrong. This is true of a well-built model, a respected consultancy, an experienced colleague, and an AI assistant. A collaborator who agrees with you, checks your arithmetic and reproduces your result has verified nothing if they are working from the same unverified number.

The protocol assumes this of itself. During the audit described in §5, the analysis produced eight errors of its own — an incorrect claim about a simulation cascade limit, a wrong stopping-power prediction, a mistaken accelerator gradient derived from confusing an energy with a length, an interpolation bug producing a false optimum, and an architectural proposal that did not survive its own arithmetic. All eight were caught by the same five rules that caught the original error, and all eight are documented.

The protocol is not a claim to be right. It is a method for finding out when you are not, before it becomes expensive.

Download: The Verification Protocol (PDF) · Version 1.1 · 30 July 2026

Dr. Mosab Hawarey · Amman, Jordan · mosab@hawarey.org · mosab.hawarey.org
Author, *Non-Equilibrium Quantum Thermodynamics of Nuclear Fusion: Resource-Theoretic Bounds, Fluctuation Theorems, and Thermodynamic Uncertainty Relations for Plasma Ignition* — AIR Journal of Engineering and Technology (2026), DOI 10.65737/AIRJET2026736.