

METHODOLOGICAL AND OPERATIONAL GUIDE

Δ-CSI Manual

Cognitive Sovereignty Index (delta) — the complete guide to the challenge-intensity index of the CounterBrain engine

Calculation version	csi-v1
Date	2026-07-06
Document	Self-sufficient on the method and the calculation — reading and verifying the formulas requires no other sources; the operational deployment details (commands, screens, calls) are in the installation documentation
Nature	Methodological and operational guide

CONTENTS

Contents

The manual is organized by audience. Part I is self-sufficient and is the common foundation; the other parts are added according to your role (the reading paths are in chapter 0.6).

0	How to use this manual
I	Understanding the Δ -CSI (for everyone)
II	Technical construction (developer)
III	Honesty and limits (investor / due diligence)
IV	Validity and validation (academic)
V	Operating procedures (operator / decision-maker)
A	Glossary
B	Three worked numerical examples
C	Complete formula reference
D	Edge cases
E	Version changelog

SECTION 0

How to use this manual

0.1 — PURPOSE

This manual explains **what the Δ -CSI is, what it measures, how it is calculated, which guarantees it offers and which it does not, and how to use it in everyday work**. It is written to be read on its own: every concept it requires is defined within these pages, with no references to other documents. Anyone who reads it from start to finish should come away able to read a score, explain it to a third party, and defend it — or criticize it — with full command of the subject.

The Δ -CSI is the metric produced by **CounterBrain**, the engine that subjects a decision to a structured adversarial challenge. The manual describes that metric; it describes the engine only as far as needed to make the number intelligible.

0.2 — SCOPE

The manual covers:

- the **meaning** of the Δ -CSI and its correct interpretation (Part I);
- the **technical construction** of the calculation, formula by formula (Part II);
- the **defenses against sycophantic output** and the declared limits (Part III);
- the **methodological foundations** of validity and the evidence-accumulation protocol (Part IV);
- the **operating procedures** for using the Δ -CSI in production (Part V);
- a **reference apparatus**: glossary, worked numerical examples, formula reference, edge cases, changelog (Appendices).

0.3 — WHAT THE Δ -CSI IS NOT

Before explaining what it is, it pays to clear the ground of three misunderstandings, because they account for nearly every misuse:

- **It is not a judgment on the merit of the decision.** It does not say whether the choice under analysis is right or wrong.
- **It is not a prediction.** It does not estimate the probability that the decision will succeed.
- **It is not a grade for the decision-maker.** It measures neither the competence of whoever made the decision nor the quality of the work done before the analysis.

The Δ -CSI measures one thing only: **how intense the adversarial challenge was** that the decision was subjected to. Part I develops this point in depth; for now it is enough to keep it in mind as the key to reading everything that follows.

0.4 — PREREQUISITES

Part I requires no technical prerequisites. For Part II and Part V, familiarity with basic programming concepts and with the notion of an environment variable is helpful. Part IV assumes fluency with the vocabulary of psychometric validity (construct validity, reproducibility, calibration), but every term is defined in the glossary regardless (Appendix A).

0.5 — TYPOGRAPHIC CONVENTIONS

- Monospaced text denotes **implementation symbols**: names of functions (`score_delta_csi`), files (`csi.py`), constants (`MIN_SAMPLE`), fields (`divergence_method`), or environment variables (`BRAIN_DIVERGENCE`). These references let developers anchor the text to the code; readers of Part I alone can ignore them without losing anything.
- The Δ -CSI **bands** (marginal, moderate, substantial) are set in roman type.
- Quotations in curly quotes (“...”) reproduce verbatim formulations from the system, or definitions.
- The **Figure N** boxes describe figures rendered graphically in this printable version of the manual.

0.6 — READING PATHS

The manual is organized by audience. Not everyone needs to read all of it: the table below sets out the minimum path for each role. **Part I is the common foundation**: anyone who uses the Δ -CSI should have read it.

AUDIENCE	PATH
Anyone who must read a score	Part I (self-sufficient)
Operator / decision-maker	Part I + Part V
Investor / due diligence	Part I + Part III
Developer / engine	Part I + Part II + Part V
Academic / standards	Part I + Part II + Part III + Part IV

PART I — FOR EVERYONE

Understanding the Δ -CSI

I.1 — A DECISION UNDER PRESSURE

A board of directors is about to approve the acquisition of a technology startup — a young artificial-intelligence company — at the multiple agreed with the sellers, that is, at a price expressed as a multiple of revenue. The goal is to accelerate the time-to-market of the flagship product: buy the technology instead of building it. The dossier is ready, the numbers add up, the mood in the room is favorable. Before the vote, the decision is submitted to CounterBrain.

What does the system do? It does not give an opinion. It does not say whether to buy, or at what price. It does something different: it puts the decision under pressure.

First, it surfaces the implicit assumptions — what the board is taking for granted without ever putting it on the record. In our example, three emerge. First: that the two technology systems are compatible enough to allow integration within twelve months. Second: that the startup's founders will accept the retention terms after the deal closes. Third: that the end customer will perceive the value of the acquired feature without additional training. None of these three sentences appears in the dossier; yet if even one of them proved false, the entire deal would change complexion.

Next, the system proposes concrete tests to try to disprove the thesis — not rhetorical objections, but executable checks. Two, in the example. First: conduct a joint technical review of the two architectures before signing; if the estimated migration effort exceeds six person-months, the thesis is falsified. Second: obtain binding commitments — deferred vesting of shares, performance-linked bonuses — from the three key engineers; if even one refuses, the thesis emerges gravely weakened.

It then poses the burden-of-proof questions: who must prove what, before proceeding. Two here as well. Who bears the burden of proving that the startup's customers are willing to migrate to the acquirer's system within eighteen months? And what is the independent estimate of the hidden integration cost — and who is contractually accountable for it?

It then constructs the plausible opposite scenario, the reverse of the thesis: the integration fails — incompatible systems, flight of key talent, dilution of company focus — and the expected value erodes. This is not a prophecy: it is a structured hypothesis against which to measure the official thesis. Finally, it cites the source used to support the challenge — a study on post-merger integrations in the technology sector, on failure rates and mitigation strategies — declaring its provenance: in this case, a document supplied by the user.

At the end of the analysis a number appears: **54.23 out of 100**, band “moderate.” What does that number say? It says that the adversarial challenge exerted moderate pressure on

the decision's assumptions: it found real material to work with — three non-trivial assumptions, two executable tests, two uncomfortable questions, an opposite scenario very distant from the thesis — but it neither demolished the deal's foundations nor merely scratched its surface. Neither a marginal challenge nor a demolition: a stress test of medium intensity.

And here lies the point that carries everything else in this manual. That 54.23 does not say whether the acquisition is a good deal. It does not say whether the board should vote yes or no. It says how hard the stress test was that the decision underwent before the vote. The difference between the two readings — judgment on the merits on one side, intensity of the verification on the other — is the subject of the next chapter. The numbers of this example are reproduced in full in Appendix B.

I.2 — WHAT IT MEASURES — AND WHAT IT DOESN'T

The quickest way to understand the Δ -CSI is to think of a sparring partner. A boxer preparing for a fight does not ask his sparring partner whether he will win: he asks him to make things hard. At the end of the training session he can say how tough the round was — how many blows he had to parry, how many times he found himself in trouble. That judgment measures the intensity of the training; it does not predict the outcome of the fight. The Δ -CSI is exactly this: it measures how tough the round was between your decision and the adversarial challenge. It does not say whether the decision will win.

Metaphor aside: the Δ -CSI measures **the intensity of the challenge** that the adversarial process exerted on the decision. Five signals feed into it: how many implicit assumptions emerged and how severe they are; how many falsification tests were proposed; how many burden-of-proof questions were posed; how far the opposite scenario diverges from the thesis; how many sources were cited. The more the challenge finds to grip — weighty assumptions, sharp tests, an opposite scenario far removed from the official thesis — the higher the number climbs.

With equal clarity, we must say what the Δ -CSI **does not measure**. It does not measure **the correctness of the decision**: a high score does not mean the choice is wrong, a low score does not mean it is right. It does not measure **the probability of success**: no element of the calculation estimates how things will turn out. It does not measure **the merit of the decision-maker**: it is not a grade for the person or group who formulated the decision, nor for the quality of the work done before the analysis. Anyone who reads the number as a report card — in either direction — commits a category error, not a mere imprecision.

Why this distinction matters so much. A category error is not fixed with more precision: it is fixed by changing the question. Asking the Δ -CSI whether the decision is right is like asking a thermometer whether the weather is nice: the instrument will answer with a number, but that number does not answer the question. No threshold, however high or low, turns a measure of intensity into a judgment on the merits. Keeping the two planes separate is not pedantry: it is the condition for the number to be useful rather than misleading.

The boxer who walks out of a punishing round has not learned whether he will win; he has learned where he is vulnerable. In the same way, a high Δ -CSI hands the decision-maker a list of weak points to examine, not a verdict. Keeping the two planes separate is the minimum requirement for using the Δ -CSI usefully.

Example. A Δ -CSI of 78 on an investment decision means the adversarial challenge found abundant material: severe assumptions, sharp tests, an opposite scenario radically different from the thesis. It does **not** mean the investment is wrong — it means there is much to look at before deciding.

Counter-example. A Δ -CSI of 12 on the same decision is **not** a green light. It may mean the decision is genuinely solid, or — and this is the more insidious case — that it was formulated so vaguely that it offered nothing precise to contradict. The low number, on its own, does not distinguish the two cases: only re-reading the decision does (chapter I.6).

I.3 — THE SCALE AND THE THREE BANDS

The Δ -CSI is expressed on a scale from 0 to 100 and is read through three bands. Below 33 the challenge was **marginal**: the adversarial process found little to grip. From 33 upward the challenge is **moderate**: the challenge produced substantial material without dislodging the framework of the thesis. From 66 upward the challenge is **substantial**: the decision's assumptions have been questioned in depth, and the decision-maker faces objections of real weight.

What each band “feels like.” The three bands are not just numeric intervals: they correspond to three different experiences for whoever reads the report.

- In the **marginal** band the report is short and toothless: few assumptions, often obvious ones; generic falsification tests; an opposite scenario that looks too much like the thesis. The correct reaction is not relief, but suspicion: *why did the challenge find so little?*
- In the **moderate** band the report has substance: non-trivial assumptions, at least a couple of genuinely executable tests, uncomfortable questions with a precise addressee, an opposite scenario that opens a different perspective. This is the band of real work: there is material to examine, but the thesis holds its frame.
- In the **substantial** band the report bites: many severe assumptions, sharp tests, an opposite scenario that inverts causes and effects. The decision-maker faces objections that cannot be ignored. It is not an order to change one's mind, but an obligation to examine each challenge one by one.

The thresholds are fixed: they do not change by sector, by decision type, or by configuration. An organization can adapt other aspects of the calculation to its own domain (we will see them in chapter I.5), but the boundaries of the three bands remain the same everywhere. This is a deliberate choice: if every installation could move the thresholds, the word “moderate” would lose its shared meaning, and the same number would tell different stories to different audiences.

The example from chapter I.1 sits at 54.23: inside the moderate band, well past the entry threshold but still far from the zone where the challenge dislodges the thesis. Appendix B places it alongside two other cases — one marginal (22) and one substantial (78) — so the three bands can be seen side by side.

The Δ -CSI scale

0–100, three reading bands

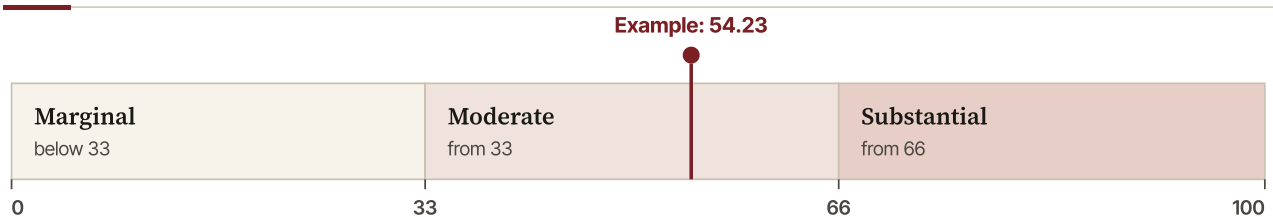


Figure 1 — The Δ -CSI scale: three fixed bands, with the example from chapter I.1 at 54.23.

I.4 — THE FIVE INGREDIENTS

The final number is born of five ingredients. No formulas are needed to understand them: words suffice. Each is presented here with the intuition that justifies it, an example of what makes it rise, and a counter-example of what does not count.

First — implicit assumptions. These are what the decision takes for granted without declaring it. How many there are counts, but so does how severe they are: an assumption that, if false, would bring down the entire thesis weighs more than a marginal one. It is the only ingredient weighted by severity as well as by count.

- *Example (counts heavily):* “the two technology systems are compatible” — if false, the entire acquisition changes complexion: a severe assumption.
- *Counter-example (counts little):* “the contract will be drafted in Italian” — true or false, it does not move the outcome: a marginal assumption.

Second — falsification tests. These are concrete experiments or checks that could disprove the thesis. Not objections in principle, but executable actions with a clear outcome criterion.

- *Example (counts):* “joint technical review of the two architectures; if the estimated migration exceeds six person-months, the thesis is falsified” — action + threshold + outcome.
- *Counter-example (does not count as falsification):* “more attention should be paid to the integration” — that is an exhortation, not a test: it has neither criterion nor outcome.

Third — burden-of-proof questions. They establish who must prove what, before proceeding. They exist to prevent the deal's most convenient claims from going without someone responsible for proving them.

- *Example (counts)*: “who bears the burden of proving that customers will migrate within eighteen months?” — it assigns a burden to a party.
- *Counter-example (does not count)*: “customers will probably migrate” — that is an optimistic assumption, not a question that assigns a burden.

Fourth — sources. These are the materials cited in support of the adversarial challenge. Each source must declare where it comes from — supplied by the user, retrieved from a search, and so on. If the system has no retrieval tools available, it does not invent any: the count simply stays at zero.

- *Example (counts)*: “study on failure rates of post-merger integrations, supplied by the user” — a source with declared provenance.
- *Counter-example (does not count)*: a claim presented as fact but without any declared source adds nothing to this ingredient.

Fifth — divergence. It measures how far the opposite scenario — the counter-scenario — moves away from the thesis. A counter-scenario that copies the thesis while changing a few words is worth little; one that truly overturns it — different causes, different consequences, different outcome — is worth a great deal.

- *Example (counts)*: thesis “the acquisition accelerates the roadmap”; counter-scenario “the integration fails, the talent leaves, focus dilutes and value erodes” — opposite causes and effects.
- *Counter-example (does not count)*: counter-scenario “the acquisition accelerates the roadmap a little less” — that is the same thesis watered down, not a reversal.

I.5 — THE TWO RULES OF THE CALCULATION

The ingredients become a number through two rules. Knowing them helps you understand why the Δ -CSI cannot be “inflated” and why it remains comparable over time.

First rule — anti-inflation. Every count-based ingredient stops growing beyond the threshold of five elements. Producing ten trivial falsifications is worth no more than five relevant ones: past that threshold, the benefit of each additional element is zero. The rule removes any incentive to manufacture volume to inflate the score. The path to a higher score runs through the substance of the challenge, not the length of the lists.

Second rule — weights that sum to one hundred. The five ingredients weigh 25, 25, 20, 10 and 20 points respectively — assumptions, falsifications, questions, sources and divergence — and the sum is always one hundred. Each weight is therefore the **maximum share** that ingredient can contribute to the score: assumptions at most 25 points, sources at most 10, and so on. The maximum Δ -CSI is 100 by construction.

An organization can customize the weights to fit its own domain — through the **configuration Packs** (Part II) — but on one iron condition: the sum must remain one hundred. Whoever increases the weight of one ingredient must take it from another. The

scale remains 0 to 100, and the numbers remain comparable over time within the same configuration. The thresholds of the three bands, by contrast, are never touched.

The five ingredients

maximum weight of each component on the score

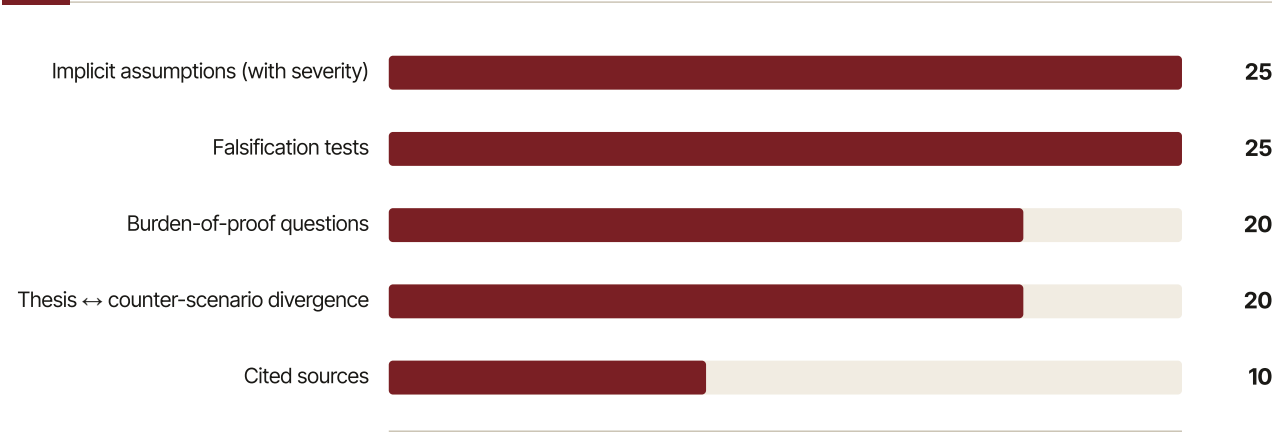


Figure 2 — The five ingredients and their maximum weight (sum of 100).

1.6 — HOW TO USE IT IN PRACTICE

The score is not a finish line: it is the start of the work. This three-step guide covers everyday use; Part V gives the detailed operational version.

Facing the score. If the score is marginal, the right question is not “did we win?” but “why did the challenge find so little?”. There are two possible explanations, and they are very different: the decision is genuinely solid and offers the challenge little grip, or it was formulated so vaguely that there was nothing precise to attack. A decision that is unassailable because it is generic is not a solid decision: it is a decision not yet formulated. It is worth rewriting it in more demanding terms and repeating the analysis. If the score is moderate or substantial, the work begins with the most severe assumptions and the proposed falsification tests: which assumptions withstand an honest examination? which tests can actually be run, before proceeding? A high score does not order you to change your mind: it orders you to look at the challenges one by one. The review can be considered complete when every severe assumption has been verified or explicitly accepted, every executable test has been run or scheduled with an owner, and every burden-of-proof question has an addressee who has answered — or a declared reason why they will not. It is worth leaving a written record of these answers alongside the decision: it is what will make the recorded outcome legible, months later.

Recording outcomes. The analysis does not end with the score. Every analyzed decision must be closed out in the decision ledger — the **Sovereign Decision Ledger** — by recording its outcome once known: correct, wrong, or mixed, also noting whether the challenge changed the decision before it was implemented. This is a step that demands discipline, because it comes weeks or months after the analysis, when attention has already moved elsewhere. But it is the step that turns a series of isolated stress tests into accumulated experience: without recorded outcomes the system can learn nothing about your context, and every score remains a snapshot without a history.

What to expect from calibration. With outcomes, the system begins to compare scores against what actually happened. It does so with two declared cautions. Below five known outcomes — decisions closed as correct, wrong, or mixed, not those still without an outcome — the numbers remain purely descriptive and the system explicitly declares that the data are insufficient: no conclusions, no extrapolation. From that threshold upward it begins to read the relationship between scores and real outcomes — always and only on your tenant: your organization, your data, your domain. Not general truths about the method, not guarantees transferable to other contexts. Anyone promising you more is promising beyond what the system declares about itself.

The learning flywheel

decision → outcome → calibration

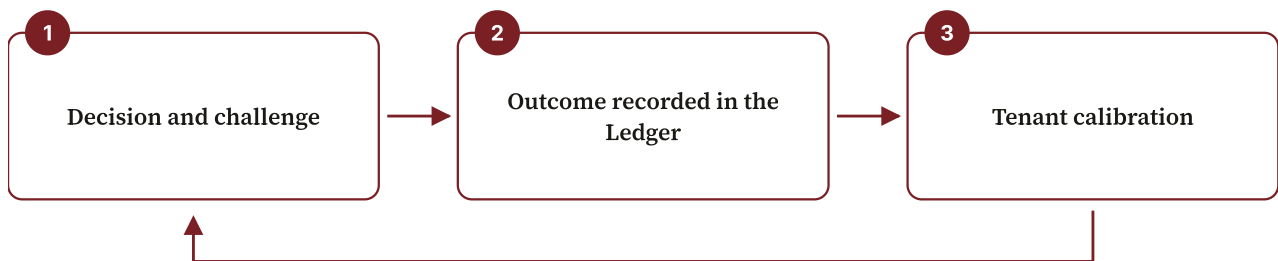


Figure 3 — The learning flywheel: decision and challenge → outcome recorded in the Ledger → calibration for the tenant.

I.7 — READING ERRORS TO AVOID

Five errors recur in reading the Δ -CSI. Knowing them in advance costs little; discovering them afterward costs misread decisions.

1. Reading it as a measure of accuracy. This is the principal category error, so much so that the system itself declares it: CounterBrain was subjected to its own method of challenge, and this is the first open risk that emerged from that self-challenge. A name that evokes “cognitive sovereignty” and a 0-to-100 scale invite reading the number as a judgment of correctness. It is not. Using the Δ -CSI to establish whether the analyzed decision is “right” means asking the number for something it does not measure — and no threshold, high or low, can answer that question.

2. Confusing it with the organizational CSI. The two indices share part of a name but operate at different levels: the Δ -CSI photographs a single analysis session, the organizational CSI the cognitive posture of an entire organization. Using them as synonyms breeds confusion about what is being measured and at what level. Chapter I.9 draws the distinction in full.

3. Believing that padding the counts raises the score. It does not work: beyond five elements, every additional assumption, falsification, question, or source is worth zero. This is the anti-inflation rule seen in chapter I.5, and it exists precisely to make artificial volume useless: the path to a higher score runs through the substance of the challenge, not the length of the lists.

4. Treating it as validated before five known outcomes. Until the decision ledger contains at least five known outcomes, the calibration numbers remain purely descriptive and the system explicitly declares that the data are insufficient to draw conclusions. Beware the mirror-image error: crossing the threshold does not turn the statistics into a validation — even above five outcomes the evidence remains descriptive and local (Part IV). Presenting the Δ -CSI as a validated measure — in a boardroom, in a due diligence, in a public document — is an error in both cases.

5. Comparing scores produced with different divergence methods. The distance between thesis and counter-scenario can be measured in two different ways, and every analysis declares which one it used. The two methods produce numbers that do not live on the same scale: placing them side by side in a direct comparison is misleading, even when they seem close. The system keeps the samples separate in calibration for exactly this reason; those who want the details will find them in Part II and Part IV.

I.8 — FREQUENTLY ASKED QUESTIONS

1. Does a high Δ -CSI mean I should change my mind?

No. It means the adversarial challenge found plenty of material to contest: severe assumptions, executable tests, a counter-scenario far from the thesis. The right response is not to reverse the decision, but to examine the assumptions one by one and ask which ones hold. The final choice remains the decision-maker's: the system applies pressure, it does not decide in your place. It is this sovereignty of the decision-maker — cognitive sovereignty — that gives the index its name.

2. Does a low Δ -CSI mean my decision is good?

No. A low score has two possible explanations, and the number alone does not distinguish them: a genuinely solid decision, offering the challenge little grip, or a decision formulated so generically that there was nothing precise to attack. Before celebrating, it pays to re-read the formulation: if it is vague, the low score is a symptom, not a compliment.

3. Is it a grade on my decision (or on me)?

No. The Δ -CSI measures the challenge process, not the merit of the decision-maker. The sparring-partner analogy applies: the judgment on the toughness of the round says nothing about the boxer's worth, and still less does it predict the outcome of the fight. The number

describes how intense the stress test was; the judgment on the decision — and on whoever made it — belongs to a plane the index does not touch.

4. Why do two analyses of the same decision give different scores?

Because the narrative part of the analysis — the texts of the assumptions, the scenarios, the questions — is produced by language models, and language models do not generate the same text twice. If the extracted elements change, the number changes. The calculation, by contrast, is always identical: given the same components, it always produces the same score. The variability lies in the generation, never in the counting.

5. Can I compare the scores of two different decisions?

With great caution. The comparison makes sense only if the two scores were produced with the same divergence measurement method and the same version of the calculation; otherwise the numbers live on different scales and placing them side by side is misleading. And even with the same method, the Δ -CSI is built to read a single analysis session, not to rank decisions against one another.

6. Who guarantees me that the system is not sycophantic?

Three structural defenses. First: if the adversarial process produces an empty challenge, the system fails visibly instead of delivering a consolation prize. Second: every source must declare its provenance, so none can materialize out of nowhere. Third: below five known outcomes, calibration declares itself insufficient rather than simulating a validation. The details of these defenses are covered in Part III.

7. What happens if there are no sources?

The system does not invent any: the field stays empty and that ingredient is worth zero points. This is the correct behavior — better a declared absence than a fabricated citation. One known limitation must be stated, however: today no explicit warning tells the reader that the synthesis was produced without sources. It is a declared weakness of the system, discussed openly in Part III.

8. Is the Δ -CSI scientifically validated?

A distinction is needed. The calculation is deterministic and reproducible by construction: anyone, with the same components, obtains the same number. Empirical validity — the number's ability to say something useful in your context — is built tenant by tenant, with recorded outcomes; and below five known outcomes the system itself forbids drawing conclusions. The full picture is in Part IV.

9. How does it differ from the CSI in the “Riarmo Cognitivo” (Cognitive Rearmament) white paper?

They operate at different levels. The Δ -CSI measures the intensity of the adversarial challenge on a single analysis session: one decision, one stress test, one number. The organizational CSI measures the cognitive posture of an entire organization, across four structural dimensions. The conceptual bridge between the two exists but is not formalized: no calibrated mapping, no analytical derivation. The distinction is explored in depth in chapter I.9.

10. Can I trust a score computed in a particular sector (healthcare, finance...)?

Yes, with the same discipline that applies everywhere. The weights of the five ingredients can be adapted to the sector through the configuration Packs — always summing to one hundred — but the reading thresholds and the logic of the calculation remain identical in every domain. And validation remains local to the tenant: the outcomes recorded in your sector speak about your sector, not about others.

11. Does the Δ -CSI rise if I add more detail to the decision?

Not directly. It rises if the challenge, having more material to work with, finds more severe assumptions, sharper tests, and a more distant opposite scenario. Adding irrelevant detail does not raise the score; formulating the decision in a more demanding, falsifiable way does — because it gives the challenge more to grip.

12. Is a score of 0 possible? And of 100?

In theory yes, at the extremes. A 0 would require the adversarial process to produce nothing on any of the five ingredients — a condition the anti-sycophancy defenses turn into a visible failure, not a delivered score (Part III). A 100 would require the maximum on all five ingredients simultaneously, including the cap of five elements on each count, assumptions all at maximum severity, and total divergence: a limiting case, not an ordinary condition.

I.9 — Δ -CSI AND ORGANIZATIONAL CSI: TWO LEVELS

In the “Riarmo Cognitivo” (Cognitive Rearmament) white paper (De Luca, Giamminola, Pollicino 2026) an index with a similar name appears: the organizational CSI. The resemblance should not mislead, because the two indices operate at different levels. The Δ -CSI measures something small and precise: the intensity of the adversarial challenge in a single analysis session, on a single decision. The organizational CSI measures something large: the cognitive sovereignty of an entire organization, assessed across four structural dimensions — how much the organization depends on algorithmic systems, how much those systems frame its decisions before they even become conscious choices, how much residual critical capacity it retains, how exposed it is to external cognitive manipulation.

The conceptual bridge between the two exists: it is reasonable to think that decisions consistently subjected to an intense adversarial challenge will, over time, end up strengthening the organization's cognitive sovereignty. But that bridge is not formalized: no calibrated mapping between the two indices exists today, nor can one be derived from the other analytically. The gap is declared and remains so. Nothing in this manual should be read as if the bridge were already built. Those who wish to explore the distinction and its consequences will find the extended treatment in Part IV.

PART II — DEVELOPER

Technical construction

This section describes how the Δ -CSI is computed by the engine. The code symbols cited (functions, constants, fields) refer to the CounterBrain engine implementation. Part I described the five ingredients in words; this part gives their formal definition, first framing it within the pipeline that produces them.

II.1 — PIPELINE OVERVIEW

The Δ -CSI does not come out of nowhere: it is the last link in a chain. Understanding where the scorer sits helps in reading correctly what it measures — and what it cannot measure.

The adversarial challenge is produced by a **three-pass** process (the three phases of the contradictor engine). In broad strokes:

- **Pass 1 — opening the challenge.** From the submitted decision the engine extracts the thesis and begins to surface the implicit assumptions and the counter-intuitive scenario.
- **Pass 2 — refinement.** The challenge is hardened: falsifications, burden-of-proof questions, consolidation of the opposing scenario. At the end of this pass the **Contradiction Floor** steps in, the check that verifies the challenge is not empty (Part III); if the floor does not hold, a **sharpened retry** is triggered.
- **Pass 3 — sources and provenance.** The engine attaches sources to the synthesis, each with its own declared provenance. If no retrieval tool is available, this pass is constrained to return an empty source list: the model does not invent citations.

Only downstream of these three passes does the **scorer** `csi.score_delta_csi` step in, receiving the five already-extracted components and turning them into a number. Here lies the crucial distinction: **the narrative part (the three passes) is produced by language models and is not deterministic; the quantitative part (the scorer) is deterministic by construction.** The Δ -CSI measures what the three passes produced, not the intrinsic quality of the arguments — a point revisited, with its consequences, in Part IV.

To de-correlate the blind spots of a single model, the three passes can run on different providers (routing controlled by `BRAIN_PASS_PROVIDERS`, with fallback to a single model if necessary; the providers actually used are recorded in `meta.models`).

II.2 — THE FIVE COMPONENTS

The Δ -CSI is computed deterministically by `csi.py:score_delta_csi` as the sum of five contributions:

COMPONENT	CORE WEIGHT	FORMULA
assumptions	25	$25 \times \min(n, 5) / 5 \times (\text{severity_avg} / 5)$
falsifications	25	$25 \times \min(n, 5) / 5$
burden_questions	20	$20 \times \min(n, 5) / 5$
sources	10	$10 \times \min(n, 5) / 5$
divergence	20	$20 \times \text{divergence (already in [0,1])}$

Sum of core weights: 100. The default weights (`DEFAULT_WEIGHTS`) are: assumptions 25 / falsifications 25 / burden_questions 20 / sources 10 / divergence 20. A Pack may override `csi_weights` as long as the sum remains 100 (`validate_weights`); the mechanism is described in chapter II.7.

Each contribution is independent of the others: the total score is their sum, with no interactions. This additivity is deliberate — it makes the number decomposable: given a score, you can always tell how many points came from each ingredient (as the examples in Appendix B show).

II.3 — THE CAP CONSTANT AND DIMINISHING MARGINAL BENEFIT

The constant `CAP = 5` governs the four count-based components (`assumptions`, `falsifications`, `burden_questions`, `sources`) through the function `_norm(n) = min(n, 5) / 5`. The result: `_norm(5) = 1.0` and `_norm(10) = 1.0`. Producing ten assumptions is not worth twice as much as five; the marginal benefit of every additional item beyond the cap is zero.

The CAP discourages inflation of the outputs. An engine that produces five dense assumptions earns the same score on the `assumptions` component as one that produces ten generic ones, given equal `severity_avg`.

The practical consequence for anyone reading a score: **a component at its maximum does not say how many items were produced, only that there are at least five.** If the raw count is needed, it must be read from the report, not inferred from the component's score.

II.4 — ASSUMPTION SEVERITY: THE ASSUMPTIONS MODULATOR

The `assumptions` component is the only one with a double multiplier. In addition to the `_norm(n)` normalization, `csi-v1` applies the factor `severity_avg / 5`.

`severity_avg` is the arithmetic mean of the `severity` fields [1-5] of each assumption in the challenge (`CsiComponents.severity_avg`). Two default conventions:

- **Missing value for an individual assumption:** it counts as 3 (neutral), so as not to penalize the accidental absence of the field.

- **Historical default (no severity declared in the challenge):** `severity_avg` is set to **5**, for backward compatibility with challenges produced before the field was introduced.

For equal `n`, a challenge with severity-5 assumptions maximizes the component (`severity_avg / 5 = 1.0`); one with severity 1 yields 20% (`severity_avg / 5 = 0.2`). In this way two challenges with the same number of assumptions but opposite severity produce different Δ -CSI values. The Δ -CSI remains the intensity of the challenge, not the quality of the decision.

Example. Three assumptions with severities {5, 5, 5} $\rightarrow$ `severity_avg = 5` $\rightarrow$ contribution `25 × 0.6 × 1.0 = 15.00`. The same three assumptions with severities {2, 2, 2} $\rightarrow$ `severity_avg = 2` $\rightarrow$ contribution `25 × 0.6 × 0.4 = 6.00`. Same number of assumptions, a contribution that drops from 15 to 6 as a function of severity.

II.5 — DIVERGENCE STRATEGIES

The divergence between thesis and counter-intuitive scenario is computed by a pluggable interface (`DivergenceStrategy` in `csi.py`). Two implementations are available.

`heuristic_divergence` (Jaccard heuristic). It operates on lowercase-normalized tokens: `divergence = 1 - Jaccard(A, B)`, where `Jaccard(A, B) = |A ∩ B| / |A ∪ B|`. Deterministic, offline, with no external dependencies. It is the guaranteed fallback in any configuration.

`embedding_cosine_divergence` (semantic cosine). It uses the embedding vectors produced by a semantic `EmbeddingProvider` (attribute `semantic=True`) injected as an `embed` function. It maps the cosine from `[-1, 1]` to `[0, 1]`: `divergence = (1 - cosine) / 2`. The model gateway does not expose an `embed` method; the cosine therefore requires a separate semantic `EmbeddingProvider`.

The conceptual difference is clear-cut: **Jaccard measures lexical distance** (different words $\rightarrow$ divergent, even when the meaning is the same), **the cosine measures semantic distance** (different meanings $\rightarrow$ divergent, even with different words). The cosine is more faithful to the construct, but it depends on an external encoder; the resulting comparability constraint is discussed in Part IV.

II.5.1 — Strategy selection: `resolve_divergence`

The `resolve_divergence` function (in `routing.py`) selects the strategy by reading the `BRAIN_DIVERGENCE` env (values: `auto` | `cosine` | `heuristic`; default `auto`) and checking for the presence of an `EmbeddingProvider` with `semantic=True`:

BRAIN_DIVERGENCE	SEMANTIC EMBEDDINGPROVIDER	SELECTED STRATEGY
auto (default)	available	semantic cosine
auto (default)	absent	Jaccard heuristic

BRAIN_DIVERGENCE	SEMANTIC EMBEDDINGPROVIDER	SELECTED STRATEGY
cosine	available	semantic cosine
cosine	absent	Jaccard heuristic (warning logged)
heuristic	any	Jaccard heuristic

Two divergence strategies

automatic selection based on configuration

Semantic cosine <i>distance in meaning</i> • Requires a semantic EmbeddingProvider • Selected when the provider is available • Faithful to the argumentative construct	Jaccard heuristic <i>lexical distance</i> • Deterministic, offline, zero dependencies • Guaranteed fallback in every configuration • Sensitive to synonyms and paraphrases
---	---

Figure 4 — Two divergence strategies, one automatic selection (resolve_divergence).

If `BRAIN_DIVERGENCE=cosine` is set without a semantic provider available, the system selects the heuristic and logs a warning — no blocking error.

II.5.2 — Encoder fail-safe

If the encoder raises an exception during the embedding call (for example because the provider is unreachable), the cosine strategy silently falls back to the Jaccard heuristic. The pipeline never crashes because of an encoder error.

The per-call fallback, besides being recorded in the log, **updates the score's divergence_method field**: the field reflects the method *actually executed* on that call, not the one nominally selected by `resolve_divergence`. A score whose divergence was computed via the Jaccard fallback is therefore labeled `heuristic` even if the selected strategy was the cosine — and it enters the right cohort in calibration. (In earlier versions of the engine the field kept the nominal value: historical records produced under that condition may contain Jaccard divergences labeled `cosine`; the limitation concerns only those records.)

II.6 — VERSIONING: FROM CSI-V0 TO CSI-V1

VERSION	ASSUMPTIONS COMPONENT	DIFFERENCE
csi-v0	$25 \times \min(n, 5) / 5$	No severity modulation
csi-v1	$25 \times \min(n, 5) / 5 \times (\text{severity_avg} / 5)$	Adds the factor $(\text{severity_avg} / 5)$

The only difference between the two versions is the factor `(severity_avg / 5)` in the `assumptions` component. All the other components remain identical. The constant `CSI_VERSION = "csi-v1"` is stamped on every score in the `csi_delta.version` field.

`csi-v0` and `csi-v1` records live on non-comparable scales, and per-cohort calibration never mixes them. The reason: for equal components, a challenge with no declared severities produces in `csi-v1` (with the default `severity_avg = 5`) the same number as `csi-v0`, but a challenge with low severities produces a lower number in `csi-v1` — the two versions are therefore not interchangeable across mixed samples.

II.7 — PACKS AND WEIGHTS: CUSTOMIZATION WITHOUT FORKS

One product principle governs all configuration: **domain specialization is expressed in configuration (the Packs), never in per-client code branches**. A Pack is a configuration file that can, among other things, redistribute the weights of the five components.

What a Pack can do. Override `csi_weights` to reflect the priorities of the domain. A sector where the documentability of sources is critical (for example clinical research or legal due diligence) can raise the weight of `sources`; a sector where implicit assumptions matter most can raise the weight of `assumptions`.

What a Pack cannot do. Break the sum constraint. The `validate_weights` function verifies, on every override, that the five weights still sum to 100; a configuration that violates the constraint is rejected. Nor can it move the **thresholds of the three bands** (marginal/moderate/substantial), which remain fixed in every domain, or alter the **calculation logic**.

Redistribution example (at a sum of one hundred). A “legal due-diligence” Pack could raise `sources` from 10 to 20 and reduce `divergence` from 20 to 10:

COMPONENT	CORE	"LEGAL" PACK
assumptions	25	25
falsifications	25	25
burden_questions	20	20
sources	10	20
divergence	20	10
Sum	100	100

The effect: within the same domain, a high score rewards the presence of documented sources more. The consequence to keep in mind: **scores produced with different weight configurations are not directly comparable** — as with versions and divergence methods,

they hold only within the same configuration (Part IV). The sum constraint nonetheless prevents customization from covertly inflating the Δ -CSI's ceiling, which remains 100.

PART III — INVESTOR / DUE DILIGENCE

Honesty and limits

This section answers the question asked by anyone examining CounterBrain in a due-diligence process: what structural defenses prevent sycophantic output? What residual weaknesses remain open? When can one speak of empirical validation? The underlying thesis is that an honest system is not one without limits, but one that **declares its limits instead of hiding them** — and that, where it can, turns them into a visible failure rather than a silent defect.

III.1 — THE THREE ANTI-SYCOPHANCY DEFENSES

CounterBrain implements three levels of structural defense against sycophantic output. They are structural in the sense that they do not depend on the model's goodwill: they are checks in the code that fire regardless of what the model produces.

1. Contradiction Floor

The `_floor_ok(challenge)` function verifies that every challenge produced meets four simultaneous criteria:

- at least 1 assumption (`assumptions` not empty);
- at least 1 falsification test (`falsification_tests` not empty);
- at least 1 burden-of-proof question (`burden_questions` not empty);
- a non-empty counter-intuitive narrative (`counter_scenario.narrative` is not an empty string).

If the floor does not hold after Pass 2, the **sharpened retry** kicks in (`sharper=True`): the model receives the explicit directive to produce a non-empty challenge meeting all four criteria. If the floor does not hold even after the retry, the pipeline raises `EngineError` — a controlled failure.

The principle is crucial in due diligence: **when the challenge is empty, CounterBrain fails visibly instead of silently degrading toward a reassuring report**. A system that, finding nothing to contradict, delivered a consolation prize anyway would be worse than useless: it would be misleading. Here the absence of a challenge becomes an error, not a result. It must be said with equal clarity that this defense sets **a floor, not a ceiling**: the floor's minimum threshold is declaredly low (it is the third of the open limits, III.2), so a poor but non-empty challenge passes the check and presents itself with a marginal score — to be read with the suspicion recommended in Part I, not as reassurance.

2. Provenance with a closed enum

Every source declared in the `sources` field must state its provenance through one of the values allowed by the closed enum:

```
web_search | internal_ledger | internal_doc | osint_module | user_provided
```

It is not possible to declare a source without stating its provenance explicitly and verifiably. If no retrieval is available (neither documents nor search tools), Pass 3 is constrained to return `sources: []`: the model cannot invent sources.

The effect in due diligence: **a source cannot materialize out of nothing**. Either it has a declared provenance among those allowed, or it does not exist in the count. This closes the door on the most insidious risk of a system that cites — the fabrication of plausible but nonexistent references.

3. Honest calibration below threshold

The `calibrate` function (`calibration.py`) computes exclusively **descriptive statistics** on historical outcomes. If a tenant's sample contains fewer than `MIN_SAMPLE = 5` outcomes with a definitive label, the interpretation is explicitly cautious: no validity conclusion is drawn, and the `sufficient` field signals to the caller that the data are not enough. The system does not extrapolate empirical validity from insufficient samples.

The effect: **the system does not simulate a validation it does not have**. Below threshold, the correct answer is not a reassuring number but the explicit statement that the data are insufficient — again, a limit made visible rather than masked.

III.2 — SELF-CHALLENGE: THE DECLARED WEAKNESSES

Consistently with the contradictor-not-oracle principle, the methodology has been subjected to its own method of challenge. What emerged is an honest list of what has already been fixed and what remains weak; the items still open are declared limits, not hidden defects.

Already fixed

- **Assumption severity.** `severity_avg` modulates the `assumptions` component as a multiplicative factor `(severity_avg / 5)`: two challenges with the same number of assumptions but opposite severities now produce distinct Δ -CSI scores.
- **Semantic divergence by default.** When a semantic `EmbeddingProvider` is available, `resolve_divergence` selects the embedding-cosine strategy — distance in meaning, not just in words — with a heuristic fallback and `divergence_method` recorded on every score.
- **Multiple independent models.** The three passes of the adversarial challenge run on different providers, to de-correlate the blind spots of a single model; if a provider is unavailable, the system falls back to a single one.

Still open

- **The Δ -CSI can be read as “correctness”.** The name and the 0–100 scale invite reading it as a judgment on the goodness of the decision, whereas it measures the intensity of the challenge. The risk is communicative and is not yet neutralized by the implementation: the Δ -CSI remains an “honest proxy”, always to be accompanied by the explicit definition of what it measures.
- **Three ingredients count quantity only.** `falsifications`, `burden_questions` and `sources` are pure counts (`_norm(n) = min(n,5)/5`): five trivial items are worth as much as five relevant ones, and argumentative quality does not enter the number. The only ingredient also weighted by severity is `assumptions`.
- **The minimum non-sycophancy threshold is low.** The `_floor_ok` check requires at least one item per field: a challenge with a single obvious assumption, a trivial test and a one-line narrative still passes it.
- **A synthesis without sources passes without an alarm.** `sources: []` passes `_floor_ok` and schema validation: the synthesis is saved and scored without any signal, and that ingredient silently contributes zero points out of ten.
- **Below five outcomes it is not empirically validated.** Until the flywheel collects at least `MIN_SAMPLE = 5` definitive outcomes per tenant, calibration remains descriptive (`sufficient=False`) and no validity conclusion is allowed.
- **Assumption severity is self-assigned by the model.** `severity_avg` — the only quality signal that enters the number — is declared by the same model that generates the challenge. The anti-inflation rule covers the counts, not the severities: a model that systematically overestimates severity inflates the heaviest component without any structural defense intercepting it.
- **Omitting severities pays more than declaring them honestly.** For backward compatibility with `csi-v0`, a challenge with no declared severity receives `severity_avg = 5` (the maximum): a higher score than an identical challenge with severities declared at 3. It is a known perverse incentive, accepted as the cost of backward compatibility; production pipelines should always declare severities.

The value of this list, in due diligence, lies not in the absence of open items — there are some — but in the fact that they are **declared by the system itself**. A vendor presenting them as already solved would be promising beyond what the system declares about itself.

III.3 — WHAT IS VALIDATED AND WHAT IS NOT

The Δ -CSI is a **deterministic, reproducible** metric: given the same input, the scorer always produces the same number. This property is verifiable and not in question.

Empirical validity is a distinct question. The Δ -CSI measures the intensity of the adversarial challenge on the declared assumptions; it is not proven that a high score corresponds to better decisions in a given tenant. That proof requires the tenant's flywheel to accumulate at least `MIN_SAMPLE = 5` definitive outcomes. Below that threshold,

`calibrate` produces only descriptive statistics and the `sufficient` field is `False`: communicating the Δ -CSI as empirically validated before reaching the threshold is an error the system flags explicitly — it is one of the declared open limits listed just above.

Once the threshold is exceeded, per-cohort calibration (`version`, `divergence_method`, `weights`) can produce a descriptive `hit_rate` — but crossing the threshold does not guarantee that `hit_rate` is populated: it remains `None` if no decision with a definitive outcome was actually changed (`calibration.py:77`). This is not causal validation: it is evidence accumulated on that tenant, not generalizable to other contexts without further analysis.

In summary, for whoever evaluates the system:

PROPERTY	STATUS	CONDITION
Determinism of the computation	Guaranteed	Always, by construction
Reproducibility of the number	Guaranteed	Given the same input components
Reproducibility of the narrative text	Not guaranteed	Depends on the language model
Per-tenant empirical validity	Buildable over time	≥ 5 definitive outcomes in the cohort
Validity generalizable across tenants	Not claimed	Requires further analysis

PART IV — ACADEMIC

Validity and validation

This section lays out the methodological foundations of the Δ-CSI for a reader assessing its theoretical soundness and its validation rigor. Points already covered in the previous sections — definition of the measured construct (Part I), formal structure of the components (Part II), anti-sycophancy defenses and declared weaknesses (Part III) — are not repeated here at length. Part IV adds the methodological why: why the Δ-CSI is built this way, what it guarantees and what it does not, how evidence about its validity is built.

IV.1 — CONSTRUCT VALIDITY

Construct validity answers a precise question: does the Δ-CSI really measure what it claims to measure?

The declared construct is **the intensity of the challenge** exerted by the contradictor engine on the decision-maker's assumptions in a single analysis session — how much pressure the adversarial challenge exerted on the thesis, not how much it actually changed the decision-maker's mind (that effect is tracked separately, by the `decision_changed` field). It is not the quality of the decision, not the probability that it will prove correct, not a judgment on the decision-maker's merit. The Δ-CSI is a proxy: an indirect quantification of the phenomenon of interest through five observable signals — assumptions detected, falsifications proposed, burden questions posed, counter-intuitive scenario produced, divergence between thesis and counter-scenario. As already stated in Part I, it is “an honest proxy”.

The operationalization is deliberate and reflects two relevant construction choices.

a) Intensity of production, not quality of arguments. Three of the five components (`falsifications`, `burden_questions`, `sources`) are pure counts normalized by `_norm(n) = min(n, 5) / 5`. The number of falsifications produced contributes to the Δ-CSI regardless of their substantive relevance. This is consistent with the declared construct — if the goal is to measure the intensity of the challenge, the volume of artifacts produced is a partial but operationally defined indicator. It is not consistent if the goal is to measure argumentative quality: that dimension is not captured. It is the same limit declared open in Part III: the pure counts are now three — `falsifications`, `burden_questions`, `sources` — because `assumptions` is now modulated by severity and no longer is one.

b) Asymmetry in the assumptions component. In `csi-v1` the `assumptions` component is modulated by `severity_avg/5`: for the same count, assumptions deemed more severe weigh more, as formalized in Part II. The other three counting components have no quality analogue. This asymmetry reflects a design choice, not a theoretically derived result: the severity of an assumption was judged to be the quality signal simplest to detect and least ambiguous to declare, and it was introduced first. Extending an analogous modulation to the other three components is a possible development direction, not a current property.

Interpretive consequences. A reader using the Δ -CSI must keep the measured construct explicit: intensity of the challenge, not accuracy. The communicative risk is among the limits declared open in Part III: the name and the 0–100 scale invite reading the number as a measure of correctness, which is not what the Δ -CSI measures. The remedy is not structural in the current implementation: it lies in educating the reader and in a communication of the Δ -CSI that always accompanies the explicit definition of the construct.

The Δ -CSI is distinct from the organizational CSI — the four-structural-dimension metric of the “Riarmo Cognitivo” white paper — in level of analysis (single decision vs. organization), in construction method (deterministic formula vs. conceptual framework) and in object. The conceptual bridge between the two is not formalized and must not be inferred from this document; the distinction is developed at length in Part I.

IV.2 — WEIGHT RATIONALE AND THE SUM CONSTRAINT

The core weights of the Δ -CSI — assumptions 25 · falsifications 25 · burden_questions 20 · sources 10 · divergence 20 — sum to 100 by construction. The constraint is enforced by the `validate_weights` function every time a Pack attempts to override `csi_weights`.

The sum to 100 has a precise interpretive consequence: each weight expresses the component's **maximum contribution share** of the total score. A component with weight 25 can contribute at most 25 points; with weight 10, at most 10. The maximum Δ -CSI is 100 by construction, not by arbitrary convention. This property makes the weights interpretable as a partition of the entire “challenge capacity” across the five operational dimensions.

The sum constraint implies that changing a weight is a zero-sum operation: every increase somewhere requires a decrease elsewhere. A Pack that raises the weight of `sources` (for example in a domain where the documentability of sources is critical) must correspondingly reduce another component. This prevents vertical customization from producing an undeclared inflation of the maximum Δ -CSI.

The default values (25/25/20/10/20) are not theoretically derived: they reflect the initial design judgment on the relative weight of the five dimensions in the general-purpose context. An empirical validation path could, in the future, estimate the relationship between the contributions of the individual components and the outcomes recorded in the ledger — an analysis that today is not defined in the calibration apparatus (the current statistics, `hit_rate` included, are not a function of the weights) and would require a ledger considerably larger than `MIN_SAMPLE`.

IV.3 — SEMANTIC AND LEXICAL

DIVERGENCE: NON-COMPARABLE MEASURES

The `divergence` component can be computed with two strategies, described in Part II: heuristic Jaccard or semantic embedding-cosine. The two strategies **measure different properties** — lexical dissimilarity the one, semantic distance the other — and produce non-comparable distributions, even though both return values in $[0, 1]$.

Jaccard measures lexical distance: it quantifies the dissimilarity between two sets of normalized tokens ($1 - |A \cap B| / |A \cup B|$). A thesis and a counter-intuitive scenario that agree in content but use different terms — synonyms, paraphrases, different languages for the same concept — come out as “divergent” under Jaccard. This is a distortion with respect to the semantic-argumentative construct of interest.

Semantic cosine measures distance in the space of embedding vectors produced by a semantic `EmbeddingProvider`. Two semantically concordant but lexically distant texts produce a high cosine (low divergence under the mapping $(1 - \text{cosine}) / 2$): behavior more faithful to the construct. Two declared trade-offs. First, the dependence on an external encoder: since the embedding provider varies by deployment, cosine scores produced by different encoders are not directly comparable — which reinforces the need for cohorts. Second, scale compression: with real encoders the cosine between two natural-language texts is rarely negative, so cosine divergence tends to stay below ~ 0.5 , while Jaccard easily reaches values close to 1 (the golden example sits at 0.9615). For the same challenge, the `divergence` component therefore systematically yields less under cosine than under Jaccard: one more reason the two cohorts must never be compared, not even by band.

This difference has direct implications for calibration. A sample of scores produced with Jaccard and a sample produced with semantic cosine are not comparable, for reasons that overlap: the divergence method (a different measured property), potentially the scorer version (different `assumptions` formula between `csi-v0` and `csi-v1`) and the weights in force (a Pack that redistributes `csi_weights` changes the maximum contribution of each component). Mixing such samples into a single calibration distribution would produce meaningless statistics: distributions generated by incommensurable measurements cannot be interpreted as samples of the same variable.

The system solves this problem with **per-cohort calibration** (`(version, divergence_method, weights)`). The `GET /v1/calibration` endpoint groups ledger records by the triple `(csi_delta.version, csi_delta.divergence_method, csi_weights)` and invokes `calibrate()` separately on each homogeneous group. Each cohort receives only samples that share the same formula for `assumptions`, the same measure for `divergence` and the same weight distribution. Records lacking `divergence_method` (produced before the field was introduced) are assigned to the `heuristic` cohort by default; records lacking `csi_weights` are assigned to the core weight configuration (25/25/20/10/20).

The methodological constraint that follows from this structure is sharp: **scores produced with different divergence strategies are not directly comparable**. A Δ -CSI of 54 produced

with Jaccard and a Δ -CSI of 54 produced with semantic cosine do not assert the same thing about the intensity of the challenge. The numerical identity is accidental, not informative.

IV.4 — EVIDENCE-ACCUMULATION PROTOCOL AND FLYWHEEL

The Δ -CSI is a metric built on theoretical premises, not empirically validated a priori. Validation is a matter of designing a process over time, not a pre-existing attribute of the formula.

The protocol is structured as a three-phase flywheel:

- **Decision.** The decision-maker submits a decision; CounterBrain produces the challenge and the Δ -CSI.
- **Outcome.** The decision-maker records the outcome in the Sovereign Decision Ledger (weeks or months later) through `outcome_label` (`corretta` | `errata` | `mista` | `ignota`) and the boolean field `decision_changed` (indicating whether the decision was actually changed as a result of the adversarial challenge).
- **Calibration.** `calibration.calibrate(samples)` analyzes the Δ -CSI distribution by bucket (marginal / moderate / substantial) and by outcome label.

The `calibrate` function produces exclusively **descriptive statistics**. It draws no causal inferences, estimates no model parameters, extrapolates no validity. Its output is an account of the distribution observed within the cohort. It must also be stated precisely what the current apparatus does *not* measure: none of the statistics produced — `hit_rate` included — statistically relates the value of the score to the outcomes; the `hit_rate` depends only on `decision_changed` and `outcome_label`. The apparatus accumulates the evidence that a future validity analysis will require; it does not, by itself, constitute a test of the Δ -CSI's validity.

The minimum threshold is `MIN_SAMPLE = 5` outcomes with a definitive label (not `ignota`) per cohort. Below that threshold, the `sufficient` field is `False` and no conclusion about the validity of the Δ -CSI is allowed by the system, as already noted in Part III. The value `MIN_SAMPLE = 5` is defined in `calibration.py`; it is a common-sense minimum operational threshold, not a threshold of statistical significance: even above threshold the evidence remains descriptive. The threshold is assessed per cohort (`version`, `divergence_method`, `weights`): when the tenant has a single cohort, “per cohort” and “per tenant” coincide.

When the threshold is exceeded, `calibrate` can produce a `hit_rate`, defined as:

HIT_RATE

changed decisions with a correct outcome / total changed decisions with a definitive outcome

The `hit_rate` can be `None` even with an above-threshold sample: if no decision with a definitive outcome was actually changed, the denominator is zero and the ratio cannot be

computed. This condition is not an anomaly: a decision-maker who never modified their choice as a result of the adversarial challenge provides no evidence for the `hit_rate`, regardless of the number of outcomes recorded.

The `hit_rate` is per-tenant evidence. It is not generalizable to other tenants or domains without further analysis; a high value in one context does not imply predictiveness in a different context. The flywheel generates the proprietary-data advantage — the per-tenant ledger grows with every decision and cannot be replicated by a third party starting from scratch — but the interpretation remains local and requires honesty about the smallness of the sample.

IV.4.1 — A worked calibration example

To make the protocol concrete, consider a tenant with **eight** closed decisions, all in the same cohort (`csi-v1`, `heuristic`, `core weights`) — thus above the `MIN_SAMPLE = 5` threshold. The table reports Δ -CSI, `outcome_label` and `decision_changed`:

#	Δ -CSI	BAND	OUTCOME_LABEL	DECISION_CHANGED
1	24	marginal	corretta	no
2	31	marginal	errata	no
3	48	moderate	corretta	yes
4	55	moderate	mista	yes
5	61	moderate	corretta	yes
6	70	substantial	errata	yes
7	78	substantial	corretta	yes
8	82	substantial	ignota	—

What `calibrate` can — and cannot — say about this sample:

- **Sufficient sample?** The definitive outcomes (not `ignota`) are seven (#1–#7). $Seven \geq 5 \rightarrow sufficient = True$. Decision #8, still `ignota`, does not contribute.
- **Distribution by band:** two marginal decisions, three moderate, three substantial (one of which still open). It is a descriptive account, not a judgment.
- **`hit_rate`:** the decisions **changed** as a result of the adversarial challenge with a **definitive** outcome are #3, #4, #5, #6, #7 (five). Of these, those with a **correct** outcome are #3, #5, #7 (three). Hence $hit_rate = 3/5 = 0.60$. Decisions #1 and #2, not changed, do not enter the denominator — consistently with the definition: the `hit_rate` measures the yield of the adversarial challenge *when it actually moved a decision*.

What **cannot** be concluded: that a high Δ -CSI “causes” better decisions (#6, substantial, has a wrong outcome), or that this 0.60 holds for another tenant. It is descriptive, local evidence on a still-small sample. The correct reading is cautious: “on this tenant, so far, when the adversarial challenge made a decision change, three times out of five the outcome was correct” — nothing more.

IV.5 — REPRODUCIBILITY

The Δ -CSI is computed by `csi.py:score_delta_csi` deterministically: given the same input — same components, same `severity_avg`, same `divergence` value — the scorer always produces the same number, regardless of the execution environment. This property does not depend on sampling, on probabilistic models or on external state.

The practical verification is in the golden example of Appendix B: score 54.23 with the values reported in the table (assumptions 3 with `severity_avg` 5.0 $\rightarrow$ 15.00; falsifications 2 $\rightarrow$ 10.00; `burden_questions` 2 $\rightarrow$ 8.00; `sources` 1 $\rightarrow$ 2.00; Jaccard divergence 0.9615 $\rightarrow$ 19.23). The reproducible Python snippet replicating that computation is included in Appendix B and serves as the reference for independent verification of the scorer implementation.

Reproducibility of the number does not imply reproducibility of the engine's narrative output. The text of the thesis, of the counter-intuitive scenario, of the assumptions and of the falsifications depends on the LLM and is not deterministic. Only the quantitative part — the Δ -CSI computed on already-extracted components — is reproducible by construction. This distinction matters for the design of regression tests: the test suite must fix the components as input to the scorer, not expect determinism from the entire narrative pipeline.

PART V — OPERATOR / DECISION-MAKER

Operating procedures

This part translates the concepts of the previous parts into actions. It describes the working cycle from the point of view of those who use CounterBrain, not of those who build it. The steps are expressed at the level of the method and of the declared interface; the exact details of commands, screens, and calls depend on the specific deployment and should be read in the installation's documentation. Wherever a step requires configuration, the text flags it explicitly.

The full cycle has five moments: **formulate and launch** an analysis, **read the report**, **record the outcome**, **configure a Pack** (when a domain adaptation is needed), **read the calibration**. The first three are the daily work; the last two are periodic.

V.1 — LAUNCHING AN ANALYSIS

Step 1 — Formulate the decision. Write the decision as a single, sharp, falsifiable statement: not “let us evaluate the acquisition”, but “proceed with the acquisition at the agreed multiple”. The more precise and demanding the formulation, the more grip it offers the adversarial challenge. A vague decision produces a weak challenge and a marginal score that is not a good sign, but a symptom (Part I, chapter I.6).

Step 2 — Attach the available context. If you have relevant documents (dossiers, studies, data), make them available to the analysis: they feed the `sources` ingredient with `user_provided` provenance and give the challenge real material to work on. In the absence of retrieval tools and documents, the system does not invent sources: the `sources` ingredient will remain at zero (correct and declared behavior).

Step 3 — Check the computation configuration. Two settings influence the number and must be known before launching:

- the **divergence strategy** in use, governed by the `BRAIN_DIVERGENCE` environment variable (`auto` | `cosine` | `heuristic`) and by the presence of a semantic `EmbeddingProvider` (Part II, chapter II.5);
- the active **Pack**, which determines the weights of the five ingredients (Part II, chapter II.7).

These two settings define the **cohort** within which the score will be comparable: scores produced with different divergence or weights cannot be compared with one another. If you do not know where to read these settings, ask whoever administers the installation: they are configuration decisions, not choices to be made at every analysis.

Step 4 — Launch. The engine runs the three passes (Part II, chapter II.1) and produces the complete challenge plus the Δ-CSI. Should the challenge turn out empty, the engine does not deliver a reassuring report: it fails visibly (`EngineError`) after the sharpened retry (Part III).

V.2 — READING A REPORT COMPONENT BY COMPONENT

A report is not read starting from the number: it is read starting from the five ingredients, and the number is understood as a consequence. Proceed in this order.

- **Implicit assumptions.** They are the most important handhold. Read them one by one and ask yourself, for each: *if this were false, would the decision change?* The high-severity assumptions are the ones to examine first — they are also the ones that weigh most in the number (the `severity_avg` factor).
- **Falsification tests.** For each one ask yourself: *can it actually be run, before proceeding? what is the outcome criterion?* A runnable test is a concrete action to put on the agenda, not a side note.
- **Burden-of-proof questions.** For each one: *who must produce this evidence, and have they produced it?* They are the way to prevent convenient claims from remaining without an owner.
- **Counter-intuitive scenario and divergence.** Read the counter-scenario and ask yourself how far it is from the thesis. A high divergence signals a radical reversal; a low one, a counter-scenario too similar to the thesis.
- **Sources.** Check the declared provenance of each one (Part III). **Beware of a known limitation:** if the list is empty, today no explicit alarm flags it for you — check it yourself. A synthesis without sources is not an error, but it must be known.

Only at this point look at the **number and the band**. It serves to direct the effort — marginal, moderate, or substantial (Part I, chapter I.3) — not to decide. The decision remains yours: the system applies pressure, it does not deliberate.

V.3 — RECORDING AN OUTCOME IN THE LEDGER

Recording the outcome is the step that feeds the system's learning, and it is also the easiest one to forget, because it comes long after the analysis.

When. When the outcome of the decision is known — weeks or months later.

What to record. Two fields:

- `outcome_label`: one of `corretta`, `errata`, `mista`, `ignota`. Until the outcome is known, it stays `ignota`; `ignota` decisions do not count toward calibration.
- `decision_changed`: a boolean value indicating whether the challenge actually caused the decision to be **modified** before it was carried out. It is the field that feeds the `hit_rate` (Part IV): without changed decisions, the `hit_rate` cannot be computed.

How to choose the label. The quality of calibration depends on the consistency of these labels, so it pays to fix the criteria *beforehand* and apply them always the same way. A minimal convention that works: **correct** if the objective declared in the decision's formulation was achieved within the expected horizon; **wrong** if it was not achieved or the scenario the decision ruled out occurred; **mixed** if the outcome is partial or the objective

was achieved at costs materially different from those expected. Label on the outcome against the declared objective, not with hindsight about what “could have been done”; if several people label, agree on the criteria in writing — two operators labeling the same outcome differently pollute the series more than a missing outcome does.

Discipline. The quality of calibration depends entirely on the constancy of this recording. A tenant that does not close its outcomes will never be able to move from the descriptive phase to a reading of the relationship between scores and outcomes. It is worth setting a delayed reminder for every decision analyzed.

V.4 — CONFIGURING A PACK

A Pack is needed when a domain has priorities that differ from the generalist ones. Configuration **requires no code changes**: it is a configuration file (Part II, chapter II.7).

Step 1 — Identify the domain priority. Example: in a legal due diligence the documentability of sources is critical; it makes sense to raise the weight of `sources`.

Step 2 — Redistribute the weights to a sum of one hundred. Every point added to one ingredient must be taken from another. The `validate_weights` function rejects any configuration whose sum is not 100. It is not possible to move the thresholds of the three bands or alter the logic of the computation: those remain identical in every domain.

Step 3 — Know that a new cohort has been opened. Changing the weights makes the scores produced by the new Pack non-comparable with those produced by the previous configuration. Calibration handles this by itself: the weights in force are recorded on every record and are part of the cohort key `(version, divergence_method, weights)`, so the two series will never be mixed. The interpretive discipline remains yours: the new series starts again from zero also for the purposes of the `MIN_SAMPLE` threshold.

V.5 — READING THE CALIBRATION OUTPUT

Calibration is obtained by querying the `GET /v1/calibration` endpoint, which groups the ledger records by cohort `(version, divergence_method, weights)` and computes the descriptive statistics separately for each one (Part IV). If you do not have direct access to the endpoint, request the report from whoever administers the installation.

What to read first — the `sufficient` field.

- If `sufficient = False`, the cohort has fewer than five definitive outcomes: **any number produced is purely descriptive and must not be communicated as validation**. This is the “under five outcomes” limit (Part III). Stop here: the only useful action is to accumulate more outcomes.
- If `sufficient = True`, you may read the distribution by band and — if populated — the `hit_rate`.

How to read the `hit_rate`. Remember the definition: changed decisions with a correct outcome over the total of changed decisions with a definitive outcome. It can be `None` even

above the threshold, if no decision has ever been changed: that is not an anomaly. And, above all: it is **evidence local to your tenant**, not a transferable truth. A high `hit_rate` on your domain says nothing about another domain.

What not to do. Do not mix cohorts. Do not compare a `hit_rate` produced with Jaccard with one produced with semantic cosine, nor scores from different versions or Packs. The system keeps cohorts separate precisely because that comparison would be meaningless.

APPENDIX A

Glossary

CSI (Cognitive Sovereignty Index)	Metric that translates the CogIA into something measurable and monitorable over time, returning the organization's cognitive posture and the distribution of epistemic authority between humans and machines.
Δ-C SI	The CounterBrain engine's operational metric that measures the intensity of the adversarial challenge in a single decision; 0–100 scale; not to be confused with the organizational CSI.
Challenge intensity	The construct measured by the Δ-C SI: how much pressure the challenge exerted on the decision's assumptions. Distinct from the correctness of the decision, from the probability of success, from the decision-maker's merit, and from the actual change of the decision (tracked by the <code>decision_changed</code> field).
Contradictor (not oracle)	Product principle whereby the system stress-tests the decision instead of issuing a verdict. Making the system "sycophantic" is considered a defect, not an improvement.
Counter-scenario (counter-intuitive scenario)	The plausible opposite scenario built by the challenge: the structured reversal of the official thesis, against which the thesis is measured.
Divergence	How far the counter-scenario moves away from the thesis. Computed lexically (Jaccard) or semantically (embedding cosine); the two methods measure different properties and produce non-comparable values.
Implicit assumption	What the decision takes for granted without stating it. The only ingredient also weighted by severity (<code>severity</code>).
Falsification test	A concrete, runnable check, with an outcome criterion, that could refute the thesis.
Burden-of-proof question	A question that establishes who must prove what before proceeding.
Source / provenance	Material cited in support of the challenge, with provenance declared from a closed set of values (<code>web_search</code> , <code>internal_ledger</code> , <code>internal_doc</code> , <code>osint_module</code> , <code>user_provided</code>).
CAP	The constant (= 5) beyond which a count ingredient stops contributing: the anti-inflation rule.
severity_avg	Arithmetic mean [1-5] of the assumptions' severities; modulates the assumptions component in <code>csi-v1</code> . Defaults to 5 when no severity is declared (backward compatibility with <code>csi-v0</code>).

Contradiction Floor	The check (<code>_floor_ok</code>) that verifies a challenge is not empty; if it does not hold, a sharpened retry is triggered and, on further failure, a controlled error.
Band	One of the three reading intervals of the Δ -CSI: marginal $[0, 33)$, moderate $[33, 66)$, substantial $[66, 100]$. Thresholds fixed in every configuration.
Cohort	Homogeneous group of scores sharing computation version, divergence method, and weight distribution (<code>version</code> , <code>divergence_method</code> , <code>weights</code>); calibration never mixes different cohorts.
Tenant	The organization that owns an installation: its data (decisions, challenges, outcomes) are isolated from every other organization's, and any calibration evidence is valid only within that perimeter.
Flywheel (learning flywheel)	The three-phase cycle decision $\rightarrow$ outcome recorded in the Ledger $\rightarrow$ calibration: each turn accumulates evidence local to the tenant.
Pack	Configuration file that adapts the system to a domain (for example by redistributing the ingredient weights to a sum of one hundred), with no per-client code branches.
Sovereign Decision Ledger	The per-tenant ledger of decisions, challenges, and outcomes; it is the data foundation that feeds calibration and constitutes the proprietary-data advantage.
hit_rate	Ratio of changed decisions with a correct outcome to the total of changed decisions with a definitive outcome; descriptive evidence local to the tenant.
MIN_SAMPLE	The threshold ($= 5$) of definitive outcomes per cohort below which calibration remains purely descriptive (<code>sufficient = False</code>).
csi-v0 / csi-v1	The two versions of the computation; they differ only in the severity factor in the assumptions component. Their records are not comparable.
System 0	Level of algorithmic processing that precedes and feeds human cognition. It sits below intuitive thinking (System 1) and analytical thinking (System 2) in dual-process theory, structuring data, options, and formulations before reasoning begins.
CogIA (Cognitive Impact Assessment)	Tool for assessing the impact of AI systems on the organization's cognitive processes. It measures three dimensions: cognitive dependence, framing effect, and critical-verification capacity.
Thinking Quadrants	Model of the four decision postures — exploratory, generative, critical, and operational — with which a decision-maker approaches interaction with AI. From it derives the diagnostic value of phenomena such as diagonal bypass and posture drift.
Cognitive lock-out	Progressive and hard-to-reverse shift of the structuring of thought toward external, opaque AI systems, with erosion of the capacity for autonomous deliberation. It is a long-term outcome, not a single error.

FIMI (Foreign Information Manipulation and Interference)	Foreign-originated information manipulation and interference: false or manipulative content injected by hostile actors to steer collective opinions and behaviors. It is monitored at the European level by the EEAS.
---	---

The Δ-C SI entry and the operational entries (challenge intensity, counter-scenario, cohort, hit_rate, Contradiction Floor, Pack, and the like) are specific to this manual; the CSI, System 0, CogIA, Thinking Quadrants, cognitive lock-out, and FIMI entries define the terms of the cognitive-security conceptual framework referenced in the text.

APPENDIX B

Three worked numerical examples

This appendix shows the Δ-CSI in action on **three cases**, one per band. The central case (the acquisition) is the reproducible golden example; the two flanking ones (finance and healthcare) are **illustrative**: their components are chosen to showcase a band, but the calculation applied is the real one, verifiable by hand.

Each contribution is obtained by applying the formulas in Appendix C. Recall the rule: count components use $\text{_norm}(n) = \min(n, 5) / 5$; **assumptions** is additionally multiplied by $\text{severity_avg} / 5$; **divergence** is already in [0,1] and is multiplied by 20.

B.1 — MARGINAL: A BARELY FALSIFIABLE FINANCIAL DECISION (Δ-CSI 22.00)

Context: “keep the current portfolio allocation for the next quarter”. A decision phrased generically, offering the adversarial challenge little to grip. Input components: 2 assumptions with $\text{severity_avg} = 2.5$; 1 falsification; 1 burden-of-proof question; no sources; heuristic divergence 0.40.

COMPONENT	N / VALUE	CALCULATION	CONTRIBUTION
assumptions	2, severity_avg 2.5	$25 \times (2/5) \times (2.5/5)$	5.00
falsifications	1	$25 \times (1/5)$	5.00
burden_questions	1	$20 \times (1/5)$	4.00
sources	0	10×0	0.00
divergence	0.40	20×0.40	8.00
Total			22.00

22.00 falls in the **marginal** band (< 33). The correct reading: not “good decision”, but “the adversarial challenge found little” — a likely symptom of a phrasing too vague, to be rewritten in more demanding terms (Part I, chapter I.6).

B.2 — MODERATE: THE ACQUISITION (GOLDEN EXAMPLE) (Δ-CSI 54.23)

Context: acquisition of a tech startup. Decision under analysis: “Proceed with the acquisition at the agreed multiple”. It is the same case told in narrative form in chapter I.1. Input components: 3 assumptions (no severity declared → $\text{severity_avg} = 5.0$), which in

`csi-v1` produces the same result as `csi-v0`); 2 falsifications; 2 burden-of-proof questions; 1 source (`user_provided`); Jaccard heuristic divergence 0.9615.

COMPONENT	N / VALUE	CALCULATION	CONTRIBUTION
assumptions	3, severity_avg 5.0	$25 \times (3/5) \times (5/5)$	15.00
falsifications	2	$25 \times (2/5)$	10.00
burden_questions	2	$20 \times (2/5)$	8.00
sources	1	$10 \times (1/5)$	2.00
divergence	0.9615	20×0.9615	19.23
Total			54.23

Textual output of the scorer (`csi.score_delta_csi`, `csi-v1`):

```
divergence = 0.9615384615384616
score      = 54.23
interpretation = Il contraddittorio ha esercitato una pressione moderata sulle
assunzioni della decisione.
```

54.23 falls in the **moderate** band [33, 66): the adversarial challenge exerted moderate pressure on the decision's assumptions.

These numbers are deterministically reproducible: given the same five components, the scorer always returns the same number. The divergence value `0.9615` is the Jaccard distance between the thesis and the counter-intuitive scenario. The following snippet replicates the calculation:

```

from brain_engine.csi import CsiComponents, score_delta_csi

# The five components of the golden example.
# severity not declared → severity_avg = 5 (historical default, backward
compatible with csi-v0).
# divergence = Jaccard distance between thesis and counter-scenario.
c = CsiComponents(
    assumptions=3,
    falsifications=2,
    burden_questions=2,
    sources=1,
    divergence=0.9615384615384616,
)
d = score_delta_csi(c)

print("score          =", d.score)          # 54.23
print("interpretation =", d.interpretation)

```

Expected output:

```

score          = 54.23
interpretation = Il contraddittorio ha esercitato una pressione moderata sulle
assunzioni della decisione.

```

The divergence can in turn be recomputed from the two texts with `from brain_engine.csi import heuristic_divergence` and `heuristic_divergence(tesi, contro_scenario)`, which returns `0.9615384615384616`.

B.3 — SUBSTANTIAL: A CLINICAL PROTOCOL PUT THROUGH THE WRINGER (Δ-CSI 78.00)

Context: “adopt the new therapeutic protocol as the ward standard”. A high-impact decision, challenged in depth. Input components: 5 assumptions with `severity_avg = 4.6`; 4 falsifications; 3 burden-of-proof questions; 3 sources; divergence 0.85.

COMPONENT	N / VALUE	CALCULATION	CONTRIBUTION
assumptions	5, severity_avg 4.6	$25 \times (5/5) \times (4.6/5)$	23.00
falsifications	4	$25 \times (4/5)$	20.00
burden_questions	3	$20 \times (3/5)$	12.00
sources	3	$10 \times (3/5)$	6.00

COMPONENT	N / VALUE	CALCULATION	CONTRIBUTION
divergence	0.85	20×0.85	17.00
Total			78.00

78.00 falls in the **substantial** band (≥ 66): the adversarial challenge questioned the decision's assumptions in depth. It is not an order to change protocol — it is the obligation to examine, one by one, five severe assumptions, four executable tests, and a very distant opposite scenario before adopting it.

NOTE ON THE REPRODUCIBILITY OF THE ILLUSTRATIVE CASES

In cases B.1 and B.3 the `severity_avg` (2.5 and 4.6) differs from the default 5; to reproduce them with the scorer it must be set explicitly, otherwise the `assumptions` component would use `severity_avg = 5` and the totals would rise to 27.00 and 80.00 respectively. The tables above report the exact calculation with the declared severities.

The three bands compared: 22 (marginal, finance) · 54.23 (moderate, acquisition) · 78 (substantial, healthcare). The same instrument, three different stress-test intensities — none of which says whether the decision was right.

APPENDIX C

Complete formula reference

A concise, self-contained reference of all calculation rules (`csi-v1`).

Count normalization: `_norm(n) = min(n, CAP) / CAP`, with `CAP = 5`. Hence `_norm(0)=0`, `_norm(1)=0.2`, ..., `_norm(5)=1.0`, `_norm(≥5)=1.0`.

The five contributions:

COMPONENT	WEIGHT	FORMULA
assumptions	25	$25 \times \text{_norm}(n) \times (\text{severity_avg} / 5)$
falsifications	25	$25 \times \text{_norm}(n)$
burden_questions	20	$20 \times \text{_norm}(n)$
sources	10	$10 \times \text{_norm}(n)$
divergence	20	$20 \times \text{divergence}$

Δ-CSI = sum of the five contributions. Theoretical maximum: 100 (weights summing to one hundred).

Severity: `severity_avg` = the [1-5] average of the assumptions' severities. A single missing severity → counts as 3. No severity declared in the challenge → `severity_avg = 5` (historical default).

Divergence:

- Jaccard (lexical): `divergence = 1 - |A ∩ B| / |A ∪ B|` over normalized tokens.
- Cosine (semantic): `divergence = (1 - cosine) / 2`, cosine mapped from [-1, 1] to [0, 1].
- Selection: `resolve_divergence` reads `BRAIN_DIVERGENCE` (`auto` | `cosine` | `heuristic`) and the availability of a semantic `EmbeddingProvider`. Guaranteed fallback: Jaccard.

Bands (fixed thresholds): marginal [0, 33) · moderate [33, 66) · substantial [66, 100].

Custom weights: a Pack may override `csi_weights` if `validate_weights` confirms sum = 100. Thresholds and logic remain unchanged.

Calibration: per cohort (`version`, `divergence_method`, `weights`); records without `csi_weights` → core weights; `sufficient = True` only with `≥ MIN_SAMPLE = 5` definitive outcomes; `hit_rate` = changed-correct / changed-definitive (may be `None`).

APPENDIX D

Edge cases

Behaviors at the extremes, useful for regression tests and for interpreting anomalous scores.

- **No sources** (`sources: []`). `sources` contribution = 0. The synthesis is valid and gets scored; no explicit alarm flags the absence (a declared limitation, Part III). To be checked manually.
- **No severity declared.** `severity_avg = 5` → `assumptions` behaves as in `csi-v0`. This is the case of the golden example.
- **All severities missing on individual assumptions.** Each absent severity counts as 3 → `severity_avg = 3` → factor `3/5 = 0.6`. Different from the “no severity declared” case (which yields 5): the distinction is deliberate.
- **Count beyond the CAP** (`n > 5`). `_norm` is already saturated at 1.0: additional items do not change the contribution. A score does not distinguish between 5 and 50 items on a count component.
- **Divergence at a band boundary.** Since `divergence` contributes up to 20 points, a small variation near a threshold (33 or 66) can change the band. The band should be read as an indication, not as truth to the last cent.
- **Same components, different versions.** With low declared severities, `csi-v0` and `csi-v1` yield different numbers; without declared severities, they coincide. Never compare them on mixed samples.
- **Encoder unreachable.** The cosine strategy falls back to Jaccard without crashing; the score records as `divergence_method` the method actually executed (`heuristic`), not the nominal one, and enters the correct cohort (Part II). Historical records produced before this fix may instead report the nominal value.
- **Empty challenge.** If `_floor_ok` does not hold even after the sharpened retry, the engine raises `EngineError`: no sycophantic score is delivered.
- **`hit_rate = None` above threshold.** Sample ≥ 5 but no decision changed → zero denominator → `hit_rate` not computable. It is not an error.
- **`ignota` outcomes.** They count neither toward the tally of definitive outcomes nor toward the `hit_rate`; a cohort made only of `ignota` remains `sufficient = False`.

APPENDIX E

Version changelog

History of the changes to the Δ-CSI calculation, with the effect of each.

csi-v1 — severity modulation of assumptions. The `assumptions` component moves from a pure count ($25 \times \text{_norm}(n)$) to a count modulated by severity ($25 \times \text{_norm}(n) \times (\text{severity_avg} / 5)$). Effect: two challenges with the same number of assumptions but opposite severity now produce distinct Δ -CSI values. It is the only formal difference from `csi-v0`; all other components remain identical. The version is stamped on every score in the `csi_delta.version` field (`CSI_VERSION = "csi-v1"`).

Semantic divergence as an available default. `resolve_divergence` automatically selects the cosine-embedding strategy when a semantic `EmbeddingProvider` is available, with a Jaccard heuristic fallback and the `divergence_method` recorded on every score. Effect: divergence can measure distance in meaning, not just in words, where the environment allows it.

Multi-model adversarial challenge. The three passes can run on different providers (routing controlled by `BRAIN_PASS_PROVIDERS`, with fallback to a single model), to de-correlate the blind spots of a single model. The providers actually used are recorded in `meta.models`.

Effective divergence method and per-weights cohorts (2026-07-05). Two consistency fixes, with no effect on the formula: (1) in case of a runtime encoder fallback, `divergence_method` records the method actually executed, not the nominal one — historical records may report the nominal value; (2) the calibration cohort key moves from `(version, divergence_method)` to `(version, divergence_method, weights)`, using the `csi_weights` recorded on each record (records without weights → core configuration). At the same time the `interpretation` string becomes “Il contraddittorio ha esercitato una pressione {marginale|moderata|sostanziale} sulle assunzioni della decisione” — literally, “The adversarial challenge exerted {marginal|moderate|substantial} pressure on the decision’s assumptions” — for adherence to the measured construct.

csi-v0 — initial version. The `assumptions` component as a pure count, without severity modulation. `csi-v0` records remain readable but are not comparable with `csi-v1` ones on mixed samples: calibration keeps them in separate cohorts.