

GUIDA METODOLOGICA E OPERATIVA

Manuale Δ -CSI

Cognitive Sovereignty Index (delta) — guida completa all'indice di intensità del contraddittorio dell'engine CounterBrain

Versione del calcolo	csi-v1
Data	2026-07-06
Documento	Autosufficiente sul metodo e sul calcolo — la lettura e la verifica delle formule non richiedono altre fonti; i dettagli operativi del deployment (comandi, schermate, chiamate) sono nella documentazione dell'installazione
Natura	Guida metodologica e operativa

INDICE

Indice

Il manuale è organizzato per pubblico. La Parte I è autosufficiente ed è il presupposto comune; le altre parti si aggiungono in funzione del ruolo (i percorsi di lettura sono nel capitolo 0.6).

O	Come usare questo manuale
I	Capire il Δ -CSI (per tutti)
II	Costruzione tecnica (sviluppatore)
III	Onestà e limiti (investitore / due-diligence)
IV	Validità e validazione (accademico)
V	Procedure operative (operatore / decisore)
A	Glossario
B	Tre esempi numerici lavorati
C	Riferimento formule completo
D	Casistiche limite
E	Changelog delle versioni

SEZIONE 0

Come usare questo manuale

0.1 — SCOPO

Questo manuale spiega **che cos'è il Δ -CSI, che cosa misura, come viene calcolato, quali garanzie offre e quali no, e come si usa nel lavoro quotidiano**. È scritto per essere letto da solo: ogni concetto necessario è definito qui dentro, senza rimandi ad altri documenti. Chi lo legge dall'inizio alla fine deve poterne uscire in grado di leggere uno score, di spiegarlo a un terzo e di difenderlo — o criticarlo — con cognizione di causa.

Il Δ -CSI è la metrica prodotta da **CounterBrain**, l'engine che sottopone una decisione a un contraddittorio strutturato. Il manuale descrive quella metrica; descrive l'engine solo quanto basta a rendere comprensibile il numero.

0.2 — AMBITO

Il manuale copre:

- il **significato** del Δ -CSI e la sua corretta interpretazione (Parte I);
- la **costruzione tecnica** del calcolo, formula per formula (Parte II);
- le **difese contro l'output compiacente** e i limiti dichiarati (Parte III);
- le **basi metodologiche** di validità e il protocollo di accumulo dell'evidenza (Parte IV);
- le **procedure operative** per usare il Δ -CSI in produzione (Parte V);
- un **apparato di riferimento**: glossario, esempi numerici lavorati, riferimento formule, casistiche limite, changelog (Appendici).

0.3 — CHE COSA IL Δ -CSI NON È

Prima ancora di spiegare che cos'è, conviene sgombrare il campo da tre malintesi, perché sono l'origine della quasi totalità degli usi scorretti:

- **Non è un giudizio sulla bontà della decisione.** Non dice se la scelta analizzata sia giusta o sbagliata.
- **Non è una previsione.** Non stima la probabilità che la decisione riesca.
- **Non è un voto al decisore.** Non misura la competenza di chi ha preso la decisione né la qualità del lavoro svolto prima dell'analisi.

Il Δ -CSI misura una cosa sola: **quanto è stato intenso il contraddittorio** a cui la decisione è stata sottoposta. La Parte I sviluppa questo punto a fondo; qui basta tenerlo presente come chiave di lettura di tutto il resto.

0.4 — PREREQUISITI

Per la Parte I non serve alcun prerequisito tecnico. Per la Parte II e la Parte V è utile familiarità con concetti di base di programmazione e con la nozione di variabile d'ambiente. La Parte IV presuppone dimestichezza con il vocabolario della validità psicometrica (validità di costrutto, riproducibilità, calibrazione), ma ogni termine è comunque definito nel glossario (Appendice A).

0.5 — CONVENZIONI TIPOGRAFICHE

- Il testo `monospaziato` indica **simboli dell'implementazione**: nomi di funzione (`score_delta_csi`), di file (`csi.py`), di costante (`MIN_SAMPLE`), di campo (`divergence_method`) o di variabile d'ambiente (`BRAIN_DIVERGENCE`). Questi riferimenti permettono a chi sviluppa di ancorare il testo al codice; chi legge solo la Parte I può ignorarli senza perdere nulla.
- Le **fasce** del Δ -CSI (marginale, moderato, sostanziale) sono scritte in tondo.
- Le citazioni fra virgolette a caporale («...») riportano formulazioni testuali del sistema o definizioni.
- I riquadri **Figura N** descrivono figure rese graficamente in questa versione stampabile del manuale.

0.6 — I PERCORSI DI LETTURA

Il manuale è organizzato per pubblico. Non tutti devono leggerlo tutto: la tabella seguente indica il percorso minimo per ciascun ruolo. La **Parte I è la base comune**: chiunque usi il Δ -CSI dovrebbe averla letta.

PUBBLICO	PERCORSO
Chiunque debba leggere uno score	Parte I (autosufficiente)
Operatore / decisore	Parte I + Parte V
Investitore / due-diligence	Parte I + Parte III
Sviluppatore / engine	Parte I + Parte II + Parte V
Accademico / standard	Parte I + Parte II + Parte III + Parte IV

PARTE I — PER TUTTI

Capire il Δ -CSI

I.1 — UNA DECISIONE SOTTO PRESSIONE

Un consiglio di amministrazione sta per approvare l'acquisizione di una startup tecnologica — una giovane azienda di intelligenza artificiale — al multiplo concordato con i venditori, cioè al prezzo espresso come multiplo dei ricavi. L'obiettivo è accelerare il time-to-market del prodotto principale: comprare la tecnologia invece di costruirla. Il dossier è pronto, i numeri tornano, il clima in sala è favorevole. Prima del voto, la decisione viene sottoposta a CounterBrain.

Cosa fa il sistema? Non dà un parere. Non dice se comprare, né a quale prezzo. Fa una cosa diversa: mette la decisione sotto pressione.

Per prima cosa fa emergere le assunzioni implicite — ciò che il consiglio sta dando per scontato senza averlo mai messo a verbale. Nel nostro esempio ne emergono tre. La prima: che i due sistemi tecnologici siano abbastanza compatibili da permettere un'integrazione entro dodici mesi. La seconda: che i fondatori della startup accettino le condizioni di permanenza dopo la chiusura dell'operazione. La terza: che il cliente finale percepisca il valore della funzionalità acquisita senza bisogno di formazione aggiuntiva. Nessuna di queste tre frasi compare nel dossier; eppure, se una sola si rivelasse falsa, l'intera operazione cambierebbe volto.

Poi il sistema propone test concreti per provare a smentire la tesi — non obiezioni retoriche, ma verifiche eseguibili. Due, nell'esempio. La prima: condurre una revisione tecnica congiunta delle due architetture prima della firma; se lo sforzo di migrazione stimato supera i sei mesi-uomo, la tesi è falsificata. La seconda: ottenere impegni vincolanti — maturazione differita delle quote, premi legati ai risultati — dai tre ingegneri chiave; se anche uno solo rifiuta, la tesi ne esce gravemente indebolita.

Quindi pone le domande d'onere della prova: chi deve dimostrare cosa, prima di procedere. Due anche qui. Chi porta la prova che i clienti della startup sono disposti a migrare sul sistema dell'acquirente entro diciotto mesi? E qual è la stima indipendente del costo nascosto di integrazione — e chi ne risponde contrattualmente?

Costruisce poi lo scenario opposto plausibile, il rovescio della tesi: l'integrazione fallisce — sistemi incompatibili, fuga dei talenti chiave, diluizione del focus aziendale — e il valore atteso si erode. Non è una profezia: è un'ipotesi strutturata contro cui misurare la tesi ufficiale. Infine cita la fonte usata a sostegno del contraddittorio — uno studio sulle integrazioni post-fusione nel settore tecnologico, sui tassi di fallimento e sulle strategie di mitigazione — dichiarandone la provenienza: in questo caso, un documento fornito dall'utente.

Alla fine dell'analisi compare un numero: **54,23 su 100**, fascia «moderato». Che cosa dice quel numero? Dice che il contraddittorio ha esercitato una pressione moderata sulle assunzioni della decisione: ha trovato materiale reale su cui lavorare — tre assunzioni non banali, due test eseguibili, due domande scomode, uno scenario opposto molto distante dalla tesi — ma non ha demolito l'impianto dell'operazione, né si è limitato a graffiarne la superficie. Né una sfida marginale né una demolizione: un collaudo di media intensità.

E qui sta il punto che regge tutto il resto di questo manuale. Quel 54,23 non dice se l'acquisizione sia un buon affare. Non dice se il consiglio debba votare sì o no. Dice quanto è stato duro il collaudo a cui la decisione è stata sottoposta prima del voto. La differenza fra le due letture — giudizio sul merito da una parte, intensità della verifica dall'altra — è l'oggetto del prossimo capitolo. I numeri di questo esempio sono riprodotti per esteso nell'Appendice B.

I.2 — COSA MISURA — E COSA NO

Il modo più rapido per capire il Δ -CSI è pensare a uno sparring partner. Un pugile che prepara un incontro non chiede al suo sparring partner se vincerà: gli chiede di metterlo in difficoltà. Alla fine dell'allenamento può dire quanto è stato duro il round — quanti colpi ha dovuto parare, quante volte è andato in difficoltà. Quel giudizio misura l'intensità dell'allenamento, non pronostica l'esito dell'incontro. Il Δ -CSI è esattamente questo: misura quanto è stato duro il round fra la vostra decisione e il contraddittorio. Non dice se la decisione vincerà.

Fuor di metafora: il Δ -CSI misura **l'intensità della sfida** che il contraddittorio ha esercitato sulla decisione. Vi concorrono cinque segnali: quante assunzioni implicite sono emerse e quanto sono gravi; quanti test di falsificazione sono stati proposti; quante domande d'onere della prova sono state poste; quanto lo scenario opposto diverge dalla tesi; quante fonti sono state citate. Più il contraddittorio trova appigli — assunzioni pesanti, test taglienti, uno scenario opposto lontanissimo dalla tesi ufficiale — più il numero sale.

Con la stessa chiarezza va detto che cosa il Δ -CSI **non misura**. Non misura **la correttezza della decisione**: un punteggio alto non significa che la scelta sia sbagliata, un punteggio basso non significa che sia giusta. Non misura **la probabilità di successo**: nessun elemento del calcolo stima come andrà a finire. Non misura **il merito del decisore**: non è un voto alla persona o al gruppo che ha formulato la decisione, né alla qualità del lavoro fatto prima dell'analisi. Chi legge il numero come una pagella — in un senso o nell'altro — commette un errore di categoria, non una semplice imprecisione.

Perché questa distinzione è così importante. Un errore di categoria non si corregge con più precisione: si corregge cambiando la domanda. Chiedere al Δ -CSI se la decisione è giusta è come chiedere a un termometro se fa bello: lo strumento risponderà un numero, ma quel numero non risponde alla domanda. Nessuna soglia, per quanto alta o bassa, trasforma una misura di intensità in un giudizio di merito. Tenere separati i due piani non è pignoleria: è la condizione perché il numero sia utile invece che fuorviante.

Il pugile che esce da un round durissimo non ha imparato se vincerà; ha imparato dove è vulnerabile. Allo stesso modo, un Δ -CSI elevato consegna al decisore un elenco di punti deboli da esaminare, non un verdetto. Tenere i due piani separati è il requisito minimo per usare il Δ -CSI in modo utile.

Esempio. Un Δ -CSI di 78 su una decisione di investimento significa che il contraddittorio ha trovato molto materiale: assunzioni gravi, test affilati, uno scenario opposto radicalmente diverso. **Non** significa che l'investimento sia sbagliato — significa che ci sono molte cose da guardare prima di decidere.

Contro-esempio. Un Δ -CSI di 12 sulla stessa decisione **non** è un via libera. Può voler dire che la decisione è davvero solida, oppure — ed è il caso più insidioso — che era formulata in modo così vago da non offrire nulla di preciso da contraddire. Il numero basso, da solo, non distingue i due casi: lo fa solo la rilettura della decisione (capitolo I.6).

I.3 — LA SCALA E LE TRE FASCE

Il Δ -CSI si esprime su una scala da 0 a 100 e si legge attraverso tre fasce. Sotto 33 la sfida è stata **marginale**: il contraddittorio ha trovato poco su cui fare presa. Da 33 in su la sfida è **moderata**: il contraddittorio ha prodotto materiale sostanzioso, senza scardinare l'impianto della tesi. Da 66 in su la sfida è **sostanziale**: le assunzioni della decisione sono state messe in discussione in profondità, e chi decide ha davanti obiezioni di peso.

Che cosa «si sente» in ciascuna fascia. Le tre fasce non sono solo intervalli numerici: corrispondono a tre esperienze diverse per chi legge il report.

- Nella fascia **marginale** il report è corto e poco mordente: poche assunzioni, spesso ovvie; test di falsificazione generici; uno scenario opposto che somiglia troppo alla tesi. La reazione corretta non è sollievo, ma sospetto: *perché il contraddittorio ha trovato così poco?*
- Nella fascia **moderata** il report ha corpo: assunzioni non banali, almeno un paio di test davvero eseguibili, domande scomode con un destinatario preciso, uno scenario opposto che apre una prospettiva diversa. È la fascia del lavoro vero: c'è materiale da esaminare, ma la tesi regge all'impianto.
- Nella fascia **sostanziale** il report morde: molte assunzioni gravi, test affilati, uno scenario opposto che ribalta cause ed effetti. Chi decide ha davanti obiezioni che non può ignorare. Non è un ordine di cambiare idea, ma un obbligo di guardare ogni sfida una per una.

Le soglie sono fisse: non cambiano per settore, per tipo di decisione né per configurazione. Un'organizzazione può adattare al proprio dominio altri aspetti del calcolo (li vedremo nel capitolo I.5), ma i confini delle tre fasce restano gli stessi ovunque. È una scelta deliberata: se ogni installazione potesse spostare le soglie, la parola «moderato» perderebbe il suo significato condiviso, e lo stesso numero racconterebbe storie diverse a interlocutori diversi.

L'esempio del capitolo I.1 si colloca a 54,23: dentro la fascia moderata, ben oltre la soglia d'ingresso ma ancora lontano dalla zona in cui il contraddittorio scardina la tesi.

Nell'Appendice B lo si affianca a due altri casi — uno marginale (22) e uno sostanziale (78) — così da vedere le tre fasce fianco a fianco.

La scala del Δ -CSI

0–100, tre fasce di lettura



Figura 1 — La scala del Δ -CSI: tre fasce fisse, con l'esempio del capitolo I.1 a 54,23.

I.4 — I CINQUE INGREDIENTI

Il numero finale nasce da cinque ingredienti. Per capirli non servono formule: bastano le parole. Ciascuno è presentato qui con l'intuizione che lo giustifica, un esempio di ciò che lo fa salire e un contro-esempio di ciò che non conta.

Primo — le assunzioni implicite. Sono ciò che la decisione dà per scontato senza dichiararlo. Contano quante sono, ma anche quanto sono gravi: un'assunzione che, se falsa, farebbe crollare l'intera tesi pesa più di una marginale. È l'unico ingrediente pesato anche per gravità, oltre che per numero.

- *Esempio (conta molto):* «i due sistemi tecnologici sono compatibili» — se falsa, l'intera acquisizione cambia volto: assunzione grave.
- *Contro-esempio (conta poco):* «il contratto sarà redatto in italiano» — vera o falsa, non sposta l'esito: assunzione marginale.

Secondo — i test di falsificazione. Sono esperimenti o verifiche concrete che potrebbero smentire la tesi. Non obiezioni di principio, ma azioni eseguibili con un criterio chiaro di esito.

- *Esempio (conta):* «revisione tecnica congiunta delle due architetture; se la migrazione stimata supera sei mesi-uomo, la tesi è falsificata» — azione + soglia + esito.
- *Contro-esempio (non conta come falsificazione):* «bisognerebbe fare più attenzione all'integrazione» — è un'esortazione, non un test: non ha né criterio né esito.

Terzo — le domande d'onere della prova. Stabiliscono chi deve dimostrare cosa, prima di procedere. Servono a impedire che le affermazioni più comode dell'operazione restino senza un responsabile che le debba provare.

- *Esempio (conta):* «chi porta la prova che i clienti migreranno entro diciotto mesi?» — assegna un onere a un soggetto.

- *Contro-esempio (non conta)*: «i clienti probabilmente migreranno» — è un'assunzione ottimistica, non una domanda che assegna un onere.

Quarto — le fonti. Sono i materiali citati a sostegno del contraddittorio. Ogni fonte deve dichiarare da dove viene — fornita dall'utente, recuperata da una ricerca, e così via. Se il sistema non dispone di strumenti di recupero, non ne inventa: il conteggio resta semplicemente a zero.

- *Esempio (conta)*: «studio sui tassi di fallimento delle integrazioni post-fusione, fornito dall'utente» — fonte con provenienza dichiarata.
- *Contro-esempio (non conta)*: un'affermazione presentata come dato di fatto ma senza alcuna fonte dichiarata non aggiunge nulla a questo ingrediente.

Quinto — la divergenza. Misura quanto lo scenario opposto — il contro-scenario — si allontana dalla tesi. Un contro-scenario che ricalca la tesi cambiando qualche parola vale poco; uno che la ribalta davvero — altre cause, altre conseguenze, altro esito — vale molto.

- *Esempio (conta)*: tesi «l'acquisizione accelera la roadmap»; contro-scenario «l'integrazione fallisce, i talenti se ne vanno, il focus si diluisce e il valore si erode» — cause ed effetti opposti.
- *Contro-esempio (non conta)*: contro-scenario «l'acquisizione accelera un po' meno la roadmap» — è la stessa tesi attenuata, non un rovescio.

I.5 — LE DUE REGOLE DEL CALCOLO

Gli ingredienti diventano numero attraverso due regole. Conoscerle serve a capire perché il Δ -CSI non si può «gonfiare» e perché resta confrontabile nel tempo.

Prima regola — l'anti-gonfiaggio. Ogni ingrediente basato sul conteggio smette di crescere oltre la soglia di cinque elementi. Produrre dieci falsificazioni banali non vale più di cinque rilevanti: oltre quella soglia, il beneficio di ogni elemento aggiuntivo è zero. La regola toglie ogni incentivo a fabbricare volume per gonfiare il punteggio. La strada per uno score più alto passa dalla sostanza della sfida, non dalla lunghezza degli elenchi.

Seconda regola — i pesi a somma cento. I cinque ingredienti pesano rispettivamente 25, 25, 20, 10 e 20 punti — assunzioni, falsificazioni, domande, fonti e divergenza — e la somma fa sempre cento. Ogni peso è dunque la **quota massima** che quell'ingrediente può contribuire allo score: le assunzioni al massimo 25 punti, le fonti al massimo 10, e così via. Il Δ -CSI massimo è 100 per costruzione.

Un'organizzazione può personalizzare i pesi per adattarli al proprio dominio — attraverso i **Pack di configurazione** (Parte II) — ma a una condizione ferrea: la somma deve restare cento. Chi aumenta il peso di un ingrediente deve toglierlo da un altro. La scala resta da 0 a 100, e i numeri restano confrontabili nel tempo all'interno della stessa configurazione. Le soglie delle tre fasce, invece, non si toccano mai.

I cinque ingredienti

peso massimo di ciascun componente sullo score

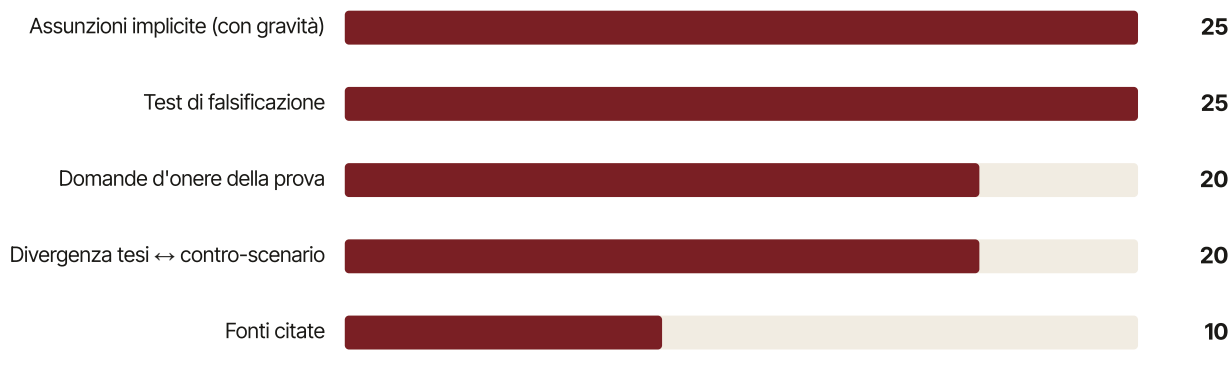


Figura 2 — I cinque ingredienti e il loro peso massimo (somma 100).

I.6 — COME USARLO IN PRATICA

Lo score non è un traguardo: è l'inizio del lavoro. Questa guida in tre passi copre l'uso quotidiano; la Parte V ne dà la versione operativa dettagliata.

Davanti allo score. Se il punteggio è marginale, la domanda giusta non è «abbiamo vinto?» ma «perché il contraddittorio ha trovato così poco?». Le spiegazioni possibili sono due, e molto diverse fra loro: la decisione è davvero solida e offre poca presa alla sfida, oppure è stata formulata in modo così vago che non c'era nulla di preciso da attaccare. Una decisione inattaccabile perché generica non è una decisione solida: è una decisione non ancora formulata. Vale la pena riscriverla in termini più impegnativi e ripetere l'analisi. Se il punteggio è moderato o sostanziale, il lavoro comincia dalle assunzioni più gravi e dai test di falsificazione proposti: quali assunzioni reggono a un esame onesto? quali test si possono eseguire davvero, prima di procedere? Un punteggio alto non ordina di cambiare idea: ordina di guardare le sfide una per una. Il riesame si può considerare concluso quando ogni assunzione grave è stata verificata o accettata esplicitamente, ogni test eseguibile è stato eseguito o messo in agenda con un responsabile, e ogni domanda d'onere ha un destinatario che ha risposto — o un motivo dichiarato per cui non risponderà. Conviene lasciare traccia scritta di queste risposte accanto alla decisione: è ciò che renderà leggibile, a distanza di mesi, il perché dell'esito registrato.

Registrare gli esiti. L'analisi non finisce con lo score. Ogni decisione analizzata va chiusa nel registro delle decisioni — il **Sovereign Decision Ledger** — registrandone l'esito quando si conosce: corretta, errata o mista, indicando anche se il contraddittorio ha fatto cambiare la decisione prima di attuarla. È un passaggio che richiede disciplina, perché arriva settimane o mesi dopo l'analisi, quando l'attenzione è già altrove. Ma è il passaggio che trasforma una serie di collaudi isolati in esperienza accumulata: senza esiti registrati il sistema non può imparare nulla sul vostro contesto, e ogni score resta una fotografia istantanea senza storia.

Cosa aspettarsi dalla calibrazione. Con gli esiti il sistema comincia a confrontare i punteggi con ciò che è realmente accaduto. Lo fa con due cautele dichiarate. Sotto i cinque esiti conosciuti — decisioni chiuse come corretta, errata o mista, non quelle ancora senza esito — i numeri restano puramente descrittivi e il sistema dichiara esplicitamente che i dati non bastano: nessuna conclusione, nessuna estrapolazione. Da quella soglia in su comincia a leggere la relazione fra i punteggi e gli esiti reali — sempre e solo sul vostro tenant: la vostra organizzazione, i vostri dati, il vostro dominio. Non verità generali sul metodo, non garanzie trasferibili ad altri contesti. Chi vi promette di più sta promettendo oltre ciò che il sistema dichiara di sé.

Il volano dell'apprendimento

decisione → esito → calibrazione



Figura 3 — Il volano dell'apprendimento: decisione e contraddittorio → esito registrato nel Ledger → calibrazione per il tenant.

I.7 — ERRORI DI LETTURA DA EVITARE

Cinque errori ricorrono nella lettura del Δ -CSI. Conoscerli in anticipo costa poco; scoprirli dopo costa decisioni interpretate male.

1. Leggerlo come misura di accuratezza. È l'errore di categoria principale, tanto che il sistema stesso lo dichiara: CounterBrain è stato sottoposto al suo stesso metodo di sfida, e questo è il primo rischio aperto emerso da quell'auto-contraddittorio. Un nome che evoca la «sovranità cognitiva» e una scala da 0 a 100 invitano a leggere il numero come un giudizio di correttezza. Non lo è. Usare il Δ -CSI per stabilire se la decisione analizzata sia «giusta» significa chiedere al numero una cosa che non misura — e nessuna soglia, alta o bassa, può rispondere a quella domanda.

2. Confonderlo con il CSI organizzativo. I due indici condividono parte del nome ma operano a livelli diversi: il Δ -CSI fotografa una singola sessione di analisi, il CSI organizzativo la postura cognitiva di un'intera organizzazione. Usarli come sinonimi genera confusione su che cosa si stia misurando e a quale livello. Il capitolo I.9 traccia la distinzione per esteso.

3. Credere che gonfiare i conteggi alzi lo score. Non funziona: oltre i cinque elementi, ogni assunzione, falsificazione, domanda o fonte aggiuntiva vale zero. È la regola anti-gonfiaggio vista nel capitolo I.5, ed esiste proprio per rendere inutile il volume artificiale: la strada per uno score più alto passa dalla sostanza della sfida, non dalla lunghezza degli elenchi.

4. Considerarlo validato prima di cinque esiti conosciuti. Finché il registro delle decisioni non contiene almeno cinque esiti conosciuti, i numeri della calibrazione restano puramente descrittivi e il sistema dichiara esplicitamente che i dati non bastano per trarre conclusioni. Attenzione all'errore speculare: superare la soglia non trasforma le statistiche in una validazione — anche sopra i cinque esiti l'evidenza resta descrittiva e locale (Parte IV). Presentare il Δ -CSI come misura validata — in un consiglio, in una due diligence, in un documento pubblico — è un errore in entrambi i casi.

5. Confrontare score prodotti con metodi di divergenza diversi. La distanza fra tesi e contro-scenario può essere misurata in due modi diversi, e ogni analisi dichiara quale ha usato. I due metodi producono numeri che non vivono sulla stessa scala: accostarli in un confronto diretto è fuorviante, anche quando sembrano vicini. Il sistema, nella calibrazione, tiene separati i campioni proprio per questo; chi vuole i dettagli li trova nella Parte II e nella Parte IV.

I.8 — DOMANDE FREQUENTI

1. Un Δ -CSI alto significa che devo cambiare idea?

No. Significa che il contraddittorio ha trovato molto materiale da sfidare: assunzioni gravi, test eseguibili, un contro-scenario distante dalla tesi. La risposta giusta non è invertire la decisione, ma guardare le assunzioni una per una e chiedersi quali reggono. La scelta finale resta del decisore: il sistema mette sotto pressione, non decide al posto vostro. È questa sovranità del decisore — la sovranità cognitiva — che dà il nome all'indice.

2. Un Δ -CSI basso significa che la mia decisione è buona?

No. Un punteggio basso ha due spiegazioni possibili, e il numero da solo non le distingue: una decisione davvero solida, che offre poca presa al contraddittorio, oppure una decisione formulata in modo così generico che non c'era nulla di preciso da attaccare. Prima di rallegrarsi conviene rileggere la formulazione: se è vaga, il punteggio basso è un sintomo, non un complimento.

3. È un voto alla mia decisione (o a me)?

No. Il Δ -CSI misura il processo di sfida, non il merito di chi decide. Vale l'analogia dello sparring partner: il giudizio sulla durezza del round non dice nulla sul valore del pugile, e ancora meno pronostica l'esito dell'incontro. Il numero descrive quanto è stato intenso il collaudo; il giudizio sulla decisione — e su chi l'ha presa — appartiene a un piano che l'indice non tocca.

4. Perché due analisi sulla stessa decisione danno score diversi?

Perché la parte narrativa dell'analisi — i testi delle assunzioni, gli scenari, le domande — è prodotta da modelli linguistici, e i modelli linguistici non generano due volte lo stesso testo. Se cambiano gli elementi estratti, cambia il numero. Il calcolo, invece, è sempre identico: dati gli stessi componenti, produce sempre lo stesso score. La variabilità sta nella generazione, mai nel conteggio.

5. Posso confrontare gli score di due decisioni diverse?

Con molta cautela. Il confronto ha senso solo se i due score sono stati prodotti con lo stesso metodo di misurazione della divergenza e con la stessa versione del calcolo; altrimenti i numeri vivono su scale diverse e accostarli è fuorviante. E anche a parità di metodo, il Δ -CSI nasce per leggere una singola sessione di analisi, non per stilare classifiche fra decisioni.

6. Chi mi garantisce che il sistema non sia compiacente?

Tre difese strutturali. Primo: se il contraddittorio produce una sfida vuota, il sistema fallisce in modo visibile invece di consegnare un contentino. Secondo: ogni fonte deve dichiarare la propria provenienza, quindi non può materializzarsi dal nulla. Terzo: sotto i cinque esiti conosciuti la calibrazione si dichiara insufficiente, invece di simulare una validazione. I dettagli di queste difese sono trattati nella Parte III.

7. Cosa succede se non ci sono fonti?

Il sistema non ne inventa: il campo resta vuoto e quell'ingrediente vale zero punti. È il comportamento corretto — meglio un'assenza dichiarata di una citazione fabbricata. Va però detto un limite noto: oggi nessun allarme esplicito segnala al lettore che la sintesi è stata prodotta senza fonti. È una debolezza dichiarata del sistema, discussa apertamente nella Parte III.

8. Il Δ -CSI è validato scientificamente?

Occorre distinguere. Il calcolo è deterministico e riproducibile per costruzione: chiunque, con gli stessi componenti, ottiene lo stesso numero. La validità empirica — la capacità del numero di dire qualcosa di utile nel vostro contesto — si costruisce invece tenant per tenant, con gli esiti registrati; e sotto i cinque esiti conosciuti il sistema stesso vieta di trarre conclusioni. Il quadro completo è nella Parte IV.

9. Che differenza c'è con il CSI del white paper «Riarmo Cognitivo»?

Operano a livelli diversi. Il Δ -CSI misura l'intensità del contraddittorio su una singola sessione di analisi: una decisione, un collaudo, un numero. Il CSI organizzativo misura la postura cognitiva di un'intera organizzazione, su quattro dimensioni strutturali. Il ponte concettuale fra i due esiste ma non è formalizzato: nessun mapping calibrato, nessuna derivazione analitica. La distinzione è approfondita nel capitolo I.9.

10. Posso fidarmi di uno score calcolato su un settore particolare (sanità, finanza...)?

Sì, con la stessa disciplina che vale ovunque. I pesi dei cinque ingredienti possono essere adattati al settore attraverso i Pack di configurazione — sempre a somma cento — ma le soglie di lettura e la logica del calcolo restano identiche in ogni dominio. E la validazione resta locale al tenant: gli esiti registrati nel vostro settore parlano del vostro settore, non di altri.

11. Il Δ -CSI cresce se aggiungo più dettagli alla decisione?

Non direttamente. Cresce se il contraddittorio, avendo più materiale su cui lavorare, trova assunzioni più gravi, test più affilati e uno scenario opposto più distante. Aggiungere dettagli irrilevanti non alza lo score; formulare la decisione in modo più impegnativo e falsificabile, sì — perché offre più presa alla sfida.

12. Uno score di 0 è possibile? E di 100?

In teoria sì, agli estremi. Uno 0 richiederebbe che il contraddittorio non produca nulla su nessuno dei cinque ingredienti — condizione che le difese anti-compiacenza rendono un fallimento visibile, non uno score consegnato (Parte III). Un 100 richiederebbe il massimo su tutti e cinque gli ingredienti contemporaneamente, incluso il tetto di cinque elementi per ciascun conteggio, assunzioni tutte alla gravità massima e divergenza totale: un caso limite, non una condizione ordinaria.

I.9 — Δ -CSI E CSI ORGANIZZATIVO: DUE LIVELLI

Nel white paper «Riarmo Cognitivo» (De Luca, Giamminola, Pollicino 2026) compare un indice dal nome simile: il CSI organizzativo. La somiglianza non deve trarre in inganno, perché i due indici operano a livelli diversi. Il Δ -CSI misura una cosa piccola e precisa: l'intensità del contraddittorio in una singola sessione di analisi, su una singola decisione. Il CSI organizzativo misura una cosa grande: la sovranità cognitiva di un'intera organizzazione, valutata su quattro dimensioni strutturali — quanto l'organizzazione dipende dai sistemi algoritmici, quanto questi ne inquadrano le decisioni prima ancora che diventino scelte consapevoli, quanta capacità critica residua conserva, quanto è esposta a manipolazione cognitiva esterna.

Il ponte concettuale fra i due esiste: è ragionevole pensare che decisioni sottoposte con costanza a un contraddittorio intenso finiscano, nel tempo, per rafforzare la sovranità cognitiva dell'organizzazione. Ma quel ponte non è formalizzato: non esiste oggi alcun mapping calibrato fra i due indici, né l'uno è derivabile dall'altro per via analitica. Il divario è dichiarato e resta tale. Nulla in questo manuale va letto come se il ponte fosse già costruito. Chi vuole approfondire la distinzione e le sue conseguenze trova la trattazione estesa nella Parte IV.

PARTE II — SVILUPPATORE

Costruzione tecnica

Questa sezione descrive come il Δ -CSI viene calcolato dall'engine. I simboli di codice citati (funzioni, costanti, campi) rimandano all'implementazione dell'engine CounterBrain. La Parte I ha descritto i cinque ingredienti a parole; questa parte ne dà la definizione formale, inquadrandola prima nella pipeline che li produce.

II.1 — PANORAMICA DELLA PIPELINE

Il Δ -CSI non nasce dal nulla: è l'ultimo anello di una catena. Capire dove si colloca lo scorer aiuta a leggere correttamente ciò che misura — e ciò che non può misurare.

Il contraddittorio è prodotto da un processo a **tre passate** (le tre fasi del motore contradictor). A grandi linee:

- **Passata 1 — apertura del contraddittorio.** Dalla decisione sottomessa il motore estrae la tesi e comincia a far emergere le assunzioni implicite e lo scenario controintuitivo.
- **Passata 2 — affinamento.** Il contraddittorio viene irrobustito: falsificazioni, domande d'onere della prova, consolidamento dello scenario opposto. Al termine di questa passata interviene il **Contradiction Floor**, il controllo che verifica che la sfida non sia vuota (Parte III); se il floor non regge, scatta un **retry affilato**.
- **Passata 3 — fonti e provenienza.** Il motore associa alla sintesi le fonti, ciascuna con la propria provenienza dichiarata. Se non è disponibile alcuno strumento di recupero, questa passata è vincolata a restituire un elenco di fonti vuoto: il modello non inventa citazioni.

Solo a valle di queste tre passate interviene lo **scorer** `csi.score_delta_csi`, che riceve i cinque componenti già estratti e li trasforma in un numero. È qui la distinzione cruciale: **la parte narrativa (le tre passate) è prodotta da modelli linguistici e non è deterministica; la parte quantitativa (lo scorer) è deterministica per costruzione.** Il Δ -CSI misura ciò che le tre passate hanno prodotto, non la qualità intrinseca degli argomenti — un punto ripreso, con le sue conseguenze, nella Parte IV.

Per de-correlare i punti ciechi di un singolo modello, le tre passate possono girare su provider diversi (routing controllato da `BRAIN_PASS_PROVIDERS`, con ripiego su un singolo modello se necessario; i provider effettivamente usati sono registrati in `meta.models`).

II.2 — I CINQUE COMPONENTI

Il Δ -CSI è calcolato deterministicamente da `csi.py:score_delta_csi` come somma di cinque contributi:

COMPONENTE	PESO CORE	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	$20 \times \text{divergence (già in [0,1])}$

Somma pesi core: 100. I pesi di default (`DEFAULT_WEIGHTS`) sono: assumptions 25 / falsifications 25 / burden_questions 20 / sources 10 / divergence 20. Un Pack può sovrascrivere `csi_weights` purché la somma rimanga 100 (`validate_weights`); il meccanismo è descritto nel capitolo II.7.

Ogni contributo è indipendente dagli altri: lo score totale è la loro somma, senza interazioni. Questa additività è deliberata — rende il numero scomponibile: dato uno score, si può sempre dire quanti punti sono arrivati da ciascun ingrediente (come mostrano gli esempi dell'Appendice B).

II.3 — LA COSTANTE CAP E IL BENEFICIO MARGINALE DECRESCENTE

La costante `CAP = 5` governa i quattro componenti di conteggio (`assumptions`, `falsifications`, `burden_questions`, `sources`) attraverso la funzione `_norm(n) = min(n, 5) / 5`. Il risultato: `_norm(5) = 1,0` e `_norm(10) = 1,0`. Produrre dieci assunzioni non vale il doppio di cinque; il beneficio marginale di ogni elemento aggiuntivo oltre il limite è zero.

Il CAP scoraggia il gonfiamento degli output. Un engine che produce cinque assunzioni dense ottiene lo stesso punteggio sul componente `assumptions` di uno che ne produce dieci generiche, a parità di `severity_avg`.

La conseguenza pratica per chi legge uno score: **un componente al suo massimo non dice quanti elementi sono stati prodotti, solo che sono almeno cinque.** Se serve conoscere il conteggio grezzo, va letto nel report, non dedotto dal punteggio del componente.

II.4 — SEVERITÀ DELLE ASSUNZIONI: IL MODULATORE DI ASSUMPTIONS

Il componente `assumptions` è l'unico ad avere un doppio moltiplicatore. Oltre alla normalizzazione `_norm(n)`, in `csi-v1` viene applicato il fattore `severity_avg / 5`.

`severity_avg` è la media aritmetica dei campi `severity` [1-5] di ogni assunzione nel challenge (`CsiComponents.severity_avg`). Due convenzioni di default:

- **Valore mancante per singola assunzione:** conta come 3 (neutro), per non penalizzare l'assenza accidentale del campo.

- **Default storico (nessuna severity dichiarata nel challenge):** `severity_avg` è impostato a 5, per retrocompatibilità con challenge prodotti prima dell'introduzione del campo.

A parità di `n`, un challenge con assunzioni a severity 5 massimizza il componente (`severity_avg / 5 = 1,0`); uno con severity 1 produce il 20% (`severity_avg / 5 = 0,2`). In questo modo due challenge con lo stesso numero di assunzioni ma gravità opposta producono Δ -CSI differenti. Il Δ -CSI rimane intensità della sfida, non qualità della decisione.

Esempio. Tre assunzioni con severità medie {5, 5, 5} $\rightarrow$ `severity_avg = 5` $\rightarrow$ contributo `25 × 0,6 × 1,0 = 15,00`. Le stesse tre assunzioni con severità {2, 2, 2} $\rightarrow$ `severity_avg = 2` $\rightarrow$ contributo `25 × 0,6 × 0,4 = 6,00`. Stesso numero di assunzioni, contributo che passa da 15 a 6 in funzione della gravità.

II.5 — STRATEGIE DI DIVERGENZA

La divergenza fra tesi e scenario controintuitivo è calcolata da un'interfaccia pluggable (`DivergenceStrategy` in `csi.py`). Sono disponibili due implementazioni.

`heuristic_divergence` (euristico Jaccard). Opera sui token normalizzati in minuscolo: `divergenza = 1 - Jaccard(A, B)`, dove `Jaccard(A, B) = |A ∩ B| / |A ∪ B|`. Deterministico, offline, senza dipendenze esterne. È il fallback garantito in qualsiasi configurazione.

`embedding_cosine_divergence` (coseno semantico). Usa i vettori di embedding prodotti da un `EmbeddingProvider` semantico (attributo `semantic=True`) iniettato come funzione `embed`. Mappa il coseno da [-1, 1] a [0, 1]: `divergenza = (1 - coseno) / 2`. Il model gateway non espone un metodo `embed`; il coseno richiede quindi un `EmbeddingProvider` semantico separato.

La differenza concettuale è netta: **Jaccard misura la distanza lessicale** (parole diverse $\rightarrow$ divergente, anche a parità di significato), **il coseno misura la distanza semantica** (significati diversi $\rightarrow$ divergente, anche con parole diverse). Il coseno è più fedele al costruito, ma dipende da un encoder esterno; il conseguente vincolo di confrontabilità è discusso nella Parte IV.

II.5.1 — Selezione della strategia: `resolve_divergence`

La funzione `resolve_divergence` (in `routing.py`) seleziona la strategia leggendo l'env `BRAIN_DIVERGENCE` (valori: `auto` | `cosine` | `heuristic`; default `auto`) e verificando la presenza di un `EmbeddingProvider` con `semantic=True`:

BRAIN_DIVERGENCE	EMBEDDINGPROVIDER SEMANTICO	STRATEGIA SELEZIONATA
auto (default)	disponibile	coseno semantico
auto (default)	assente	euristico Jaccard
cosine	disponibile	coseno semantico

BRAIN_DIVERGENCE	EMBEDDINGPROVIDER SEMANTICO	STRATEGIA SELEZIONATA
cosine	assente	euristico Jaccard (warning loggato)
heuristic	qualsiasi	euristico Jaccard

Due strategie di divergenza

selezione automatica in base alla configurazione

Coseno semantico <i>distanza di significato</i> • Richiede un EmbeddingProvider semantico • Scelto quando il provider è disponibile • Fedele al costruito argomentativo	Euristica Jaccard <i>distanza lessicale</i> • Deterministica, offline, zero dipendenze • Fallback garantito in ogni configurazione • Sensibile a sinonimi e parafrasi
---	---

Figura 4 — Due strategie di divergenza, una selezione automatica (resolve_divergence).

Se `BRAIN_DIVERGENCE=cosine` è impostato senza un provider semantico disponibile, il sistema seleziona l'euristico e logga un warning — nessun errore bloccante.

II.5.2 — Fail-safe encoder

Se l'encoder solleva un'eccezione durante la chiamata di embedding (ad esempio per un provider irraggiungibile), la strategia coseno ripiega silenziosamente sull'euristico Jaccard. La pipeline non va mai in crash per un errore dell'encoder.

Il fallback per-chiamata, oltre a essere registrato nel log, **aggiorna il campo `divergence_method` dello score**: il campo riflette il metodo *effettivamente eseguito* su quella chiamata, non quello nominalmente selezionato da `resolve_divergence`. Uno score la cui divergenza è stata calcolata in ripiego Jaccard viene quindi etichettato `heuristic` anche se la strategia selezionata era il coseno — ed entra nella coorte giusta in calibrazione. (Nelle versioni precedenti dell'engine il campo restava il valore nominale: i record storici prodotti in quella condizione possono contenere divergenze Jaccard etichettate `cosine`; il limite riguarda solo quei record.)

II.6 — VERSIONING: DA CSI-V0 A CSI-V1

VERSIONE	COMPONENTE ASSUMPTIONS	DIFFERENZA
csi-v0	$25 \times \min(n, 5) / 5$	Nessuna modulazione per severità
csi-v1	$25 \times \min(n, 5) / 5 \times (\text{severity_avg} / 5)$	Aggiunge il fattore (<code>severity_avg</code> / 5)

L'unica differenza fra le due versioni è il fattore `(severity_avg / 5)` nel componente `assumptions`. Tutti gli altri componenti restano identici. La costante `CSI_VERSION = "csi-v1"` è stampata su ogni score nel campo `csi_delta.version`.

I record `csi-v0` e `csi-v1` vivono su scale non confrontabili e la calibrazione per coorte non li mescola mai. La ragione: a parità di componenti, un challenge senza severità dichiarate produce in `csi-v1` (con `severity_avg = 5` di default) lo stesso numero di `csi-v0`, ma un challenge con severità basse produce in `csi-v1` un numero più basso — le due versioni non sono quindi intercambiabili su campioni misti.

II.7 — PACK E PESI: PERSONALIZZAZIONE SENZA FORK

Un principio di prodotto governa tutta la configurazione: **la specializzazione di dominio si esprime in configurazione (i Pack), mai in rami di codice per singolo cliente**. Un Pack è un file di configurazione che può, fra le altre cose, ridistribuire i pesi dei cinque componenti.

Cosa un Pack può fare. Sovrascrivere `csi_weights` per riflettere le priorità del dominio. Un settore in cui la documentabilità delle fonti è critica (per esempio la ricerca clinica o la due-diligence legale) può alzare il peso di `sources`; un settore in cui contano soprattutto le assunzioni implicite può alzare il peso di `assumptions`.

Cosa un Pack non può fare. Rompere il vincolo di somma. La funzione `validate_weights` verifica, a ogni sovrascrittura, che i cinque pesi sommino ancora a 100; una configurazione che non rispetta il vincolo viene rifiutata. Non può inoltre spostare le **soglie delle tre fasce** (marginale/moderato/sostanziale), che restano fisse in ogni dominio, né alterare la **logica del calcolo**.

Esempio di ridistribuzione (a somma cento). Un Pack «due-diligence legale» potrebbe portare `sources` da 10 a 20 e ridurre `divergence` da 20 a 10:

COMPONENTE	CORE	PACK «LEGALE»
<code>assumptions</code>	25	25
<code>falsifications</code>	25	25
<code>burden_questions</code>	20	20
<code>sources</code>	10	20
<code>divergence</code>	20	10
Somma	100	100

L'effetto: nello stesso dominio, uno score alto premia di più la presenza di fonti documentate. La conseguenza da tenere presente: **score prodotti con configurazioni di pesi diverse non sono direttamente confrontabili** — valgono, come per le versioni e i

metodi di divergenza, solo all'interno della stessa configurazione (Parte IV). Il vincolo di somma impedisce comunque che la personalizzazione gonfi in modo occulto il tetto massimo del Δ -CSI, che resta 100.

PARTE III — INVESTITORE / DUE-DILIGENCE

Onestà e limiti

Questa sezione risponde alla domanda di chi esamina CounterBrain in un processo di due-diligence: quali difese strutturali prevengono output compiacente? Quali debolezze residue restano aperte? Quando si può parlare di validazione empirica? La tesi di fondo è che un sistema onesto non è quello che non ha limiti, ma quello che **dichiara i propri limiti invece di nasconderli** — e che, dove può, li rende un fallimento visibile anziché un difetto silenzioso.

III.1 — LE TRE DIFESE ANTI-COMPIACENZA

CounterBrain implementa tre livelli di difesa strutturale contro l'output compiacente. Sono strutturali nel senso che non dipendono dalla buona volontà del modello: sono controlli nel codice che scattano indipendentemente da ciò che il modello produce.

1. Contradiction Floor

La funzione `_floor_ok(challenge)` verifica che ogni sfida prodotta superi quattro criteri simultanei:

- almeno 1 assunzione (`assumptions` non vuoto);
- almeno 1 test di falsificazione (`falsification_tests` non vuoto);
- almeno 1 domanda d'onere della prova (`burden_questions` non vuoto);
- narrativa controintuitiva non vuota (`counter_scenario.narrative` non è stringa vuota).

Se il floor non regge dopo la Passata 2, scatta il **retry affilato** (`sharper=True`): il modello riceve la direttiva esplicita di produrre una sfida non vuota rispettando tutti e quattro i criteri. Se il floor non regge neanche dopo il retry, la pipeline lancia `EngineError` — fallimento controllato.

Il principio è cruciale in sede di due-diligence: **quando la sfida è vuota, CounterBrain fallisce visibilmente invece di degradare silenziosamente verso un report rassicurante**. Un sistema che, non trovando nulla da contraddire, consegnasse comunque un contentino sarebbe peggio che inutile: sarebbe fuorviante. Qui l'assenza di sfida diventa un errore, non un risultato. Va detto con la stessa chiarezza che questa difesa fissa **un pavimento, non un soffitto**: la soglia minima del floor è dichiaratamente bassa (è il terzo dei limiti aperti, III.2), quindi un contraddittorio povero ma non vuoto supera il controllo e si presenta con uno score marginale — che va letto con il sospetto raccomandato nella Parte I, non come rassicurazione.

2. Provenienza con enum chiuso

Ogni fonte dichiarata nel campo `sources` deve indicare la propria provenienza attraverso uno dei valori ammessi dall'enum chiuso:

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

Non è possibile dichiarare una fonte senza indicarne la provenienza in modo esplicito e verificabile. Se non è disponibile retrieval (né documenti né strumenti di ricerca), la Passata 3 è vincolata a restituire `sources: []`: il modello non può inventare fonti.

L'effetto in due-diligence: **una fonte non può materializzarsi dal nulla**. O ha una provenienza dichiarata fra quelle ammesse, o non esiste nel conteggio. Questo chiude la porta al rischio più insidioso di un sistema che cita — la fabbricazione di riferimenti plausibili ma inesistenti.

3. Calibrazione onesta sotto soglia

La funzione `calibrate` (`calibration.py`) computa esclusivamente **statistiche descrittive** sugli esiti storici. Se il campione di un tenant è inferiore a `MIN_SAMPLE = 5` esiti con outcome definitivo, l'interpretazione è esplicitamente prudente: nessuna conclusione di validità viene tratta, e il campo `sufficient` segnala al chiamante che i dati non bastano. Il sistema non extrapola validità empirica da campioni insufficienti.

L'effetto: **il sistema non simula una validazione che non ha**. Sotto soglia, la risposta corretta non è un numero rassicurante, ma la dichiarazione esplicita che i dati sono insufficienti — di nuovo, un limite reso visibile invece che mascherato.

III.2 — AUTO-CONTRADDITTORIO: LE DEBOLEZZE DICHIARATE

Coerentemente con il principio contraddittore-non-oracolo, la metodologia è stata sottoposta al suo stesso metodo di sfida. Ne è uscito un elenco onesto di ciò che è già stato corretto e di ciò che resta debole; le voci ancora aperte sono limiti dichiarati, non difetti nascosti.

Già corretto

- **Gravità delle assunzioni.** `severity_avg` modula il componente `assumptions` come fattore moltiplicativo (`severity_avg / 5`): due challenge con lo stesso numero di assunzioni ma gravità opposta producono ora Δ -CSI distinti.
- **Divergenza semantica di default.** Quando è disponibile un `EmbeddingProvider` semantico, `resolve_divergence` sceglie la strategia coseno-embedding — distanza di significato, non solo di parole — con ripiego euristico e `divergence_method` registrato su ogni score.
- **Più modelli indipendenti.** Le tre passate del contraddittorio girano su provider diversi, per de-correlare i punti ciechi di un singolo modello; se un provider non è disponibile, il sistema ripiega su uno solo.

Ancora aperto

- **Il Δ-CSI può essere letto come «correttezza».** Il nome e la scala 0–100 inducono a leggerlo come giudizio sulla bontà della decisione, mentre misura l'intensità della sfida. Il rischio è comunicativo e non è ancora neutralizzato dall'implementazione: il Δ-CSI resta un «proxy onesto», da accompagnare sempre con la definizione esplicita di ciò che misura.
- **Tre ingredienti contano solo la quantità.** `falsifications`, `burden_questions` e `sources` sono conteggi puri (`_norm(n) = min(n,5)/5`): cinque elementi banali valgono quanto cinque rilevanti, e la qualità argomentativa non entra nel numero. L'unico ingrediente pesato anche per gravità è `assumptions`.
- **La soglia minima di non-compiacenza è bassa.** Il controllo `_floor_ok` richiede almeno un elemento per campo: un challenge con una sola assunzione ovvia, un test banale e una narrativa di una riga lo supera comunque.
- **Una sintesi senza fonti passa senza allarme.** `sources: []` supera `_floor_ok` e la validazione dello schema: la sintesi viene salvata e scorata senza alcun segnale, e quell'ingrediente contribuisce zero punti su dieci in silenzio.
- **Sotto cinque esiti non è validato empiricamente.** Finché il flywheel non raccoglie almeno `MIN_SAMPLE = 5` esiti definitivi per tenant, la calibrazione resta descrittiva (`sufficient=False`) e nessuna conclusione di validità è consentita.
- **La gravità delle assunzioni è auto-attribuita dal modello.** `severity_avg` — l'unico segnale di qualità che entra nel numero — è dichiarato dallo stesso modello che genera il challenge. La regola anti-gonfiaggio copre i conteggi, non le severità: un modello che sovrastima sistematicamente la gravità gonfia il componente più pesante senza che alcuna difesa strutturale lo intercetti.
- **Omettere le severità rende più che dichiararle oneste.** Per retrocompatibilità con `csi-v0`, un challenge senza alcuna severità dichiarata riceve `severity_avg = 5` (il massimo): score più alto di un challenge identico con severità dichiarate a 3. È un incentivo perverso noto, accettato come costo della retrocompatibilità; le pipeline di produzione dovrebbero dichiarare sempre le severità.

Il valore di questo elenco, in due-diligence, non sta nell'assenza di voci aperte — che ci sono — ma nel fatto che siano **dichiarate dal sistema stesso**. Un fornitore che le presentasse come già risolte starebbe promettendo oltre ciò che il sistema dichiara di sé.

III.3 — COSA È VALIDATO E COSA NO

Il Δ-CSI è una metrica **deterministica e riproducibile**: a parità di input, lo scorer produce sempre lo stesso numero. Questa proprietà è verificabile e non è in discussione.

La **validità empirica** è una questione distinta. Il Δ-CSI misura l'intensità del contraddittorio sulle assunzioni dichiarate; non è dimostrato che uno score alto corrisponda a decisioni migliori in un dato tenant. La dimostrazione richiede che il flywheel del tenant accumuli almeno `MIN_SAMPLE = 5` esiti definitivi. Sotto quella soglia, `calibrate` produce solo statistiche descrittive e il campo `sufficient` è `False`: comunicare

il Δ -CSI come validato empiricamente prima di raggiungere la soglia è un errore che il sistema segnala esplicitamente — è uno dei limiti dichiarati aperti poco sopra.

Quando la soglia è superata, la calibrazione per coorte (`version`, `divergence_method`, `pesi`) può produrre un `hit_rate` descrittivo — ma il superamento della soglia non garantisce che `hit_rate` sia valorizzato: resta `None` se nessuna decisione con esito definitivo risulta effettivamente cambiata (`calibration.py:77`). Non si tratta di validazione causale: è evidenza accumulata su quel tenant, non generalizzabile ad altri contesti senza ulteriore analisi.

In sintesi, per chi valuta il sistema:

PROPRIETÀ	STATO	CONDIZIONE
Determinismo del calcolo	Garantito	Sempre, per costruzione
Riproducibilità del numero	Garantita	A parità di componenti in ingresso
Riproducibilità del testo narrativo	Non garantita	Dipende dal modello linguistico
Validità empirica per-tenant	Costruibile nel tempo	≥ 5 esiti definitivi nella coorte
Validità generalizzabile fra tenant	Non affermata	Richiede analisi ulteriore

PARTE IV — ACCADEMICO

Validità e validazione

Questa sezione espone le basi metodologiche del Δ -CSI per un lettore che ne valuta la fondatezza teorica e il rigore di validazione. I punti già trattati nelle sezioni precedenti — definizione del costrutto misurato (Parte I), struttura formale dei componenti (Parte II), difese anti-compiacenza e debolezze dichiarate (Parte III) — non sono ripetuti qui per esteso. La Parte IV aggiunge il perché metodologico: perché il Δ -CSI è costruito così, cosa garantisce e cosa no, come si costruisce evidenza sulla sua validità.

IV.1 — VALIDITÀ DI COSTRUTTO

La validità di costrutto risponde a una domanda precisa: il Δ -CSI misura davvero ciò che dichiara di misurare?

Il costrutto dichiarato è **l'intensità della sfida** esercitata dal motore contradictor sulle assunzioni del decisore in una singola sessione di analisi — quanta pressione il contraddittorio ha esercitato sulla tesi, non quanto abbia effettivamente cambiato l'opinione del decisore (quell'effetto è tracciato a parte, dal campo `decision_changed`). Non è la qualità della decisione, non è la probabilità che essa si riveli corretta, non è un giudizio di merito sul decisore. Il Δ -CSI è un proxy: una quantificazione indiretta del fenomeno d'interesse attraverso cinque segnali osservabili — assunzioni rilevate, falsificazioni proposte, domande d'onere poste, scenario controintuitivo prodotto, divergenza fra tesi e contro-scenario. Come già indicato in Parte I, è «un proxy onesto».

L'operazionalizzazione è deliberata e riflette due scelte di costruzione rilevanti.

a) Intensità della produzione, non qualità degli argomenti. Tre dei cinque componenti (`falsifications`, `burden_questions`, `sources`) sono conteggi puri normalizzati da `_norm(n) = min(n,5)/5`. Il numero di falsificazioni prodotte contribuisce al Δ -CSI indipendentemente dalla loro rilevanza sostanziale. Questo è coerente con il costrutto dichiarato — se si vuol misurare l'intensità della sfida, il volume di artefatti prodotti è un indicatore parziale ma operazionalmente definito. Non è coerente se si vuol misurare la qualità argomentativa: quella dimensione non è catturata. È lo stesso limite dichiarato aperto nella Parte III: i conteggi puri restano tre — `falsifications`, `burden_questions`, `sources` — perché `assumptions` è ora modulato per gravità e non lo è più.

b) Asimmetria nel componente `assumptions`. In `csi-v1` il componente `assumptions` è modulato da `severity_avg/5`: a parità di conteggio, assunzioni ritenute più gravi pesano di più, come formalizzato nella Parte II. Gli altri tre componenti di conteggio non hanno un analogo di qualità. Questa asimmetria riflette una scelta progettuale, non un risultato teoricamente derivato: si è ritenuto che la gravità di un'assunzione fosse il segnale di qualità più semplice da rilevare e meno ambiguo da dichiarare, e lo si è introdotto per primo. Estendere una modulazione analoga agli altri tre componenti è una direzione di sviluppo possibile, non una proprietà attuale.

Conseguenze interpretative. Il lettore che usa il Δ -CSI deve tenere esplicito il costrutto misurato: intensità della sfida, non accuratezza. Il rischio comunicativo è tra i limiti dichiarati aperti nella Parte III: il nome e la scala 0–100 invitano a leggere il numero come misura di correttezza, che non è ciò che il Δ -CSI misura. Il rimedio non è strutturale nell'implementazione attuale: è nell'educazione del lettore e in una comunicazione del Δ -CSI che accompagni sempre la definizione esplicita del costrutto.

Il Δ -CSI si distingue dal CSI organizzativo — metrica a quattro dimensioni strutturali del white paper «Riarmo Cognitivo» — per livello di analisi (singola decisione vs. organizzazione), per metodo di costruzione (formula deterministica vs. framework concettuale) e per oggetto. Il ponte concettuale fra i due non è formalizzato e non deve essere inferito da questo documento; la distinzione è sviluppata per esteso nella Parte I.

IV.2 — RAZIONALE DEI PESI E VINCOLO DI SOMMA

I pesi core del Δ -CSI — `assumptions` 25 · `falsifications` 25 · `burden_questions` 20 · `sources` 10 · `divergence` 20 — sommano a 100 per costruzione. Il vincolo è applicato dalla funzione `validate_weights` ogni volta che un Pack tenta di sovrascrivere `csi_weights`.

La somma a 100 ha una conseguenza interpretativa precisa: ogni peso esprime la **quota di contributo massimo** del componente allo score totale. Un componente con peso 25 può contribuire al massimo 25 punti; con peso 10, al massimo 10. Il Δ -CSI massimo è 100 per costruzione, non per convenzione arbitraria. Questa proprietà rende i pesi interpretabili come ripartizione dell'intera «capacità di sfida» fra le cinque dimensioni operative.

Il vincolo di somma implica che modificare un peso sia un'operazione a somma zero: ogni incremento da qualche parte richiede un decremento altrove. Un Pack che aumenta il peso di `sources` (ad esempio in un dominio dove la documentabilità delle fonti è critica) deve ridurre corrispondentemente un altro componente. Questo impedisce che la personalizzazione verticale produca un'inflazione non dichiarata del Δ -CSI massimo.

I valori di default (25/25/20/10/20) non sono teoricamente derivati: riflettono il giudizio progettuale iniziale sul peso relativo delle cinque dimensioni nel contesto general-purpose. Un percorso di validazione empirica potrebbe, in futuro, stimare la relazione fra i contributi dei singoli componenti e gli esiti registrati nel ledger — un'analisi che oggi non è definita nell'apparato di calibrazione (le statistiche attuali, `hit_rate` incluso, non sono funzione dei pesi) e che richiederebbe un ledger di dimensioni sensibilmente superiori a `MIN_SAMPLE`.

IV.3 — DIVERGENZA SEMANTICA E LESSICALE: MISURE NON CONFRONTABILI

Il componente `divergence` può essere calcolato con due strategie, descritte nella Parte II: euristico Jaccard o coseno-embedding semantico. Le due strategie **misurano proprietà diverse** — dissimilarità lessicale l'una, distanza semantica l'altra — e producono distribuzioni non confrontabili, pur restituendo entrambe valori in $[0, 1]$.

Jaccard misura la distanza lessicale: quantifica la dissimilarità fra due insiemi di token normalizzati ($1 - |A \cap B| / |A \cup B|$). Una tesi e uno scenario controintuitivo che concordano nel contenuto ma usano termini diversi — sinonimi, parafrasi, lingue diverse per lo stesso concetto — risultano «divergenti» in Jaccard. Questa è una distorsione rispetto al costruito semantico-argomentativo di interesse.

Coseno semantico misura la distanza nello spazio dei vettori di embedding prodotti da un `EmbeddingProvider` semantico. Due testi semanticamente concordi ma lessicalmente distanti producono un coseno alto (bassa divergenza nel mapping $(1 - \text{coseno}) / 2$): comportamento più fedele al costruito. Due trade-off dichiarati. Primo, la dipendenza da un encoder esterno: poiché il provider di embedding varia per deployment, score coseno prodotti da encoder diversi non sono direttamente confrontabili — il che rafforza la necessità delle coorti. Secondo, la compressione di scala: con encoder reali il coseno fra due testi in linguaggio naturale è raramente negativo, quindi la divergenza coseno tende a restare sotto ~0,5, mentre la Jaccard raggiunge facilmente valori prossimi a 1 (l'esempio golden è a 0,9615). A parità di challenge, il componente `divergence` rende quindi sistematicamente meno sotto coseno che sotto Jaccard: un motivo in più per cui le due coorti non vanno mai confrontate, nemmeno per fascia.

Questa differenza ha implicazioni dirette sulla calibrazione. Un campione di score prodotti con Jaccard e un campione prodotti con coseno semantico non sono confrontabili per ragioni che si sovrappongono: il metodo di divergenza (proprietà misurata diversa), potenzialmente la versione dello scorer (formula `assumptions` diversa fra `csi-v0` e `csi-v1`) e i pesi in vigore (un Pack che ridistribuisce `csi_weights` cambia il contributo massimo di ciascun componente). Mescolare tali campioni in una stessa distribuzione di calibrazione produrrebbe statistiche prive di significato: distribuzioni generate da misurazioni incommensurabili non possono essere interpretate come campioni della stessa variabile.

Il sistema risolve questo problema con la **calibrazione per coorte** (`version, divergence_method, pesi`). L'endpoint `GET /v1/calibration` raggruppa i record del ledger per tripla (`csi_delta.version, csi_delta.divergence_method, csi_weights`) e invoca `calibrate()` separatamente su ciascun gruppo omogeneo. Ogni coorte riceve solo campioni che condividono la stessa formula per `assumptions`, la stessa misura per `divergence` e la stessa distribuzione dei pesi. Record privi di `divergence_method` (prodotti prima dell'introduzione del campo) sono assegnati alla coorte `heuristic` per default; record privi di `csi_weights` sono assegnati alla configurazione dei pesi core (25/25/20/10/20).

Il vincolo metodologico che discende da questa struttura è netto: **score prodotti con strategie di divergenza diverse non sono confrontabili direttamente**. Un Δ -CSI di 54 prodotto con Jaccard e un Δ -CSI di 54 prodotto con coseno semantico non affermano la stessa cosa sull'intensità della sfida. L'identità numerica è accidentale, non informativa.

IV.4 — PROTOCOLLO DI ACCUMULO DELL'EVIDENZA E FLYWHEEL

Il Δ -CSI è una metrica costruita su premesse teoriche, non validata empiricamente a priori. La validazione è una questione di progettazione del processo nel tempo, non un attributo preesistente della formula.

Il protocollo è strutturato come un flywheel a tre fasi:

- **Decisione.** Il decisore sottomette una decisione; CounterBrain produce il challenge e il Δ -CSI.
- **Esito.** Il decisore registra l'esito nel Sovereign Decision Ledger (settimane o mesi dopo) attraverso `outcome_label` (`corretta` | `errata` | `mista` | `ignota`) e il campo booleano `decision_changed` (indica se la decisione è stata effettivamente modificata a seguito del contraddittorio).
- **Calibrazione.** `calibration.calibrate(samples)` analizza la distribuzione Δ -CSI per bucket (marginale / moderato / sostanziale) e per label di esito.

La funzione `calibrate` produce esclusivamente **statistiche descrittive**. Non produce inferenze causali, non stima parametri di un modello, non extrapola validità. Il suo output è un resoconto della distribuzione osservata all'interno della coorte. Va dichiarato con precisione anche che cosa l'apparato attuale *non* misura: nessuna delle statistiche prodotte — `hit_rate` incluso — mette in relazione statistica il valore dello score con gli esiti; il `hit_rate` dipende solo da `decision_changed` e `outcome_label`. L'apparato accumula l'evidenza che una futura analisi di validità richiederà; non costituisce, da solo, un test della validità del Δ -CSI.

La soglia minima è `MIN_SAMPLE = 5` esiti con outcome definitivo (non `ignota`) per coorte. Sotto quella soglia, il campo `sufficient` è `False` e nessuna conclusione sulla validità del Δ -CSI è consentita dal sistema, come già osservato nella Parte III. Il valore `MIN_SAMPLE = 5` è definito in `calibration.py`; è una soglia operativa minima di buon senso, non una soglia di significatività statistica: anche sopra soglia l'evidenza resta descrittiva. La soglia è valutata per coorte (`version`, `divergence_method`, `pesi`): quando il tenant ha un'unica coorte, «per coorte» e «per tenant» coincidono.

Quando la soglia è superata, `calibrate` può produrre un `hit_rate`, definito come:

HIT_RATE

decisioni cambiate con esito corretto / totale decisioni cambiate con esito definitivo

Il `hit_rate` può essere `None` anche con campione sopra soglia: se nessuna decisione con esito definitivo risulta effettivamente cambiata, il denominatore è zero e il rapporto non è

calcolabile. Questa condizione non è un'anomalia: un decisore che non ha mai modificato la propria scelta a seguito del contraddittorio non fornisce evidenza per il `hit_rate`, indipendentemente dal numero di esiti registrati.

Il `hit_rate` è evidenza per-tenant. Non è generalizzabile ad altri tenant o domini senza analisi ulteriore; un valore alto in un contesto non implica predittività in un contesto diverso. Il flywheel genera il vantaggio del dato proprietario — il ledger per-tenant cresce con ogni decisione e non è replicabile da un terzo partendo da zero — ma l'interpretazione rimane locale e richiede onestà sulla limitatezza del campione.

IV.4.1 — Un esempio di calibrazione lavorato

Per rendere concreto il protocollo, si consideri un tenant con **otto** decisioni chiuse, tutte nella stessa coorte (`csi-v1`, `heuristic`, `pesi core`) — quindi sopra la soglia `MIN_SAMPLE = 5`. La tabella riporta `Δ-CSI`, `outcome_label` e `decision_changed`:

#	Δ-CSI	FASCIA	OUTCOME_LABEL	DECISION_CHANGED
1	24	marginale	corretta	no
2	31	marginale	errata	no
3	48	moderato	corretta	sì
4	55	moderato	mista	sì
5	61	moderato	corretta	sì
6	70	sostanziale	errata	sì
7	78	sostanziale	corretta	sì
8	82	sostanziale	ignota	—

Che cosa può — e non può — dire `calibrate` su questo campione:

- **Campione sufficiente?** Gli esiti definitivi (non `ignota`) sono sette (#1–#7). Sette $\geq 5 \rightarrow$ `sufficient = True`. La decisione #8, ancora `ignota`, non concorre.
- **Distribuzione per fascia:** due decisioni marginali, tre moderate, tre sostanziali (di cui una ancora aperta). È un resoconto descrittivo, non un giudizio.
- **`hit_rate`:** le decisioni **cambiate** a seguito del contraddittorio con esito **definitivo** sono #3, #4, #5, #6, #7 (cinque). Di queste, quelle con esito **corretto** sono #3, #5, #7 (tre). Quindi `hit_rate = 3/5 = 0,60`. Le decisioni #1 e #2, non cambiate, non entrano nel denominatore — coerentemente con la definizione: il `hit_rate` misura la resa del contraddittorio *quando ha effettivamente spostato una decisione*.

Cosa **non** si può concludere: che un Δ -CSI alto «causi» decisioni migliori (la #6, sostanziale, ha esito errato), né che questo 0,60 valga per un altro tenant. È evidenza descrittiva, locale, su un campione ancora piccolo. La lettura corretta è prudente: «su questo tenant, finora, quando il contraddittorio ha fatto cambiare una decisione, tre volte su cinque l'esito è stato corretto» — nulla di più.

IV.5 — RIPRODUCIBILITÀ

Il Δ -CSI è calcolato da `csi.py:score_delta_csi` in modo deterministico: a parità di input — stessi componenti, stessa `severity_avg`, stesso valore di `divergence` — lo scorer produce sempre il medesimo numero, indipendentemente dall'ambiente di esecuzione. Questa proprietà non dipende da campionamento, da modelli probabilistici né da stato esterno.

La verifica pratica è nell'esempio golden dell'Appendice B: score 54,23 con i valori riportati nella tabella (assumptions 3 con `severity_avg` 5,0 $\rightarrow$ 15,00; falsifications 2 $\rightarrow$ 10,00; `burden_questions` 2 $\rightarrow$ 8,00; `sources` 1 $\rightarrow$ 2,00; `divergence` Jaccard 0,9615 $\rightarrow$ 19,23). Lo snippet Python riproducibile che replica tale calcolo è incluso nell'Appendice B e costituisce il riferimento per la verifica indipendente dell'implementazione dello scorer.

La riproducibilità del numero non implica riproducibilità dell'output narrativo dell'engine. Il testo della tesi, dello scenario controintuitivo, delle assunzioni e delle falsificazioni dipende dal modello LLM e non è deterministico. Solo la parte quantitativa — il Δ -CSI calcolato su componenti già estratti — è riproducibile per costruzione. Questa distinzione è rilevante per la progettazione di test di regressione: la suite di test deve fissare i componenti come input allo scorer, non aspettarsi determinismo dall'intera pipeline narrativa.

PARTE V — OPERATORE / DECISORE

Procedure operative

Questa parte traduce i concetti delle parti precedenti in azioni. Descrive il ciclo di lavoro dal punto di vista di chi usa CounterBrain, non di chi lo costruisce. I passaggi sono espressi al livello del metodo e dell'interfaccia dichiarata; i dettagli esatti di comandi, schermate e chiamate dipendono dal deployment specifico e vanno letti nella documentazione dell'installazione. Dove un passaggio richiede una configurazione, il testo lo segnala esplicitamente.

Il ciclo completo ha cinque momenti: **formulare e lanciare** un'analisi, **leggere il report**, **registrare l'esito**, **configurare un Pack** (quando serve un adattamento di dominio), **leggere la calibrazione**. I primi tre sono il lavoro quotidiano; gli ultimi due sono periodici.

V.1 — LANCIARE UN'ANALISI

Passo 1 — Formulare la decisione. Scrivete la decisione come un'affermazione singola, netta e falsificabile: non «valutiamo l'acquisizione», ma «procedere con l'acquisizione al multiplo concordato». Più la formulazione è precisa e impegnativa, più offre presa al contraddittorio. Una decisione vaga produce un contraddittorio debole e uno score marginale che non è un buon segno, ma un sintomo (Parte I, capitolo I.6).

Passo 2 — Allegare il contesto disponibile. Se avete documenti pertinenti (dossier, studi, dati), metteteli a disposizione dell'analisi: alimentano l'ingrediente `sources` con provenienza `user_provided` e danno al contraddittorio materiale reale su cui lavorare. In assenza di strumenti di recupero e di documenti, il sistema non inventa fonti: l'ingrediente `sources` resterà a zero (comportamento corretto e dichiarato).

Passo 3 — Verificare la configurazione di calcolo. Due impostazioni influenzano il numero e vanno conosciute prima di lanciare:

- la **strategia di divergenza** in uso, governata dalla variabile d'ambiente `BRAIN_DIVERGENCE` (`auto` | `cosine` | `heuristic`) e dalla presenza di un `EmbeddingProvider` semantico (Parte II, capitolo II.5);
- il **Pack** attivo, che determina i pesi dei cinque ingredienti (Parte II, capitolo II.7).

Queste due impostazioni definiscono la **coorte** entro cui lo score sarà confrontabile: score prodotti con divergenza o pesi diversi non si confrontano fra loro. Se non sapete dove leggere queste impostazioni, chiedete a chi amministra l'installazione: sono decisioni di configurazione, non scelte da fare a ogni analisi.

Passo 4 — Lanciare. L'engine esegue le tre passate (Parte II, capitolo II.1) e produce il challenge completo più il Δ -CSI. Se il contraddittorio risultasse vuoto, l'engine non consegna un report rassicurante: fallisce in modo visibile (`EngineError`) dopo il retry affilato (Parte III).

V.2 — LEGGERE UN REPORT COMPONENTE PER COMPONENTE

Un report non si legge partendo dal numero: si legge partendo dai cinque ingredienti, e il numero si capisce di conseguenza. Procedete in quest'ordine.

- **Assunzioni implicite.** Sono l'appiglio più importante. Leggetele una per una e chiedetevi, per ciascuna: *se questa fosse falsa, la decisione cambierebbe?* Le assunzioni con gravità alta sono quelle da esaminare per prime — sono anche quelle che pesano di più nel numero (fattore `severity_avg`).
- **Test di falsificazione.** Per ciascuno chiedetevi: *si può eseguire davvero, prima di procedere? qual è il criterio d'esito?* Un test eseguibile è un'azione concreta da mettere in agenda, non una nota a margine.
- **Domande d'onere della prova.** Per ciascuna: *chi deve portare questa prova, e l'ha portata?* Sono il modo per impedire che le affermazioni comode restino senza un responsabile.
- **Scenario controintuitivo e divergenza.** Leggete il contro-scenario e chiedetevi quanto è distante dalla tesi. Una divergenza alta segnala un rovescio radicale; una bassa, un contro-scenario troppo simile alla tesi.
- **Fonti.** Verificate la provenienza dichiarata di ciascuna (Parte III). **Attenzione a un limite noto:** se l'elenco è vuoto, oggi nessun allarme esplicito ve lo segnala — controllatelo voi. Una sintesi senza fonti non è un errore, ma va saputa.

Solo a questo punto guardate il **numero e la fascia**. Serve a orientare lo sforzo — marginale, moderato o sostanziale (Parte I, capitolo I.3) — non a decidere. La decisione resta vostra: il sistema mette sotto pressione, non delibera.

V.3 — REGISTRARE UN ESITO NEL LEDGER

La registrazione dell'esito è il passaggio che alimenta l'apprendimento del sistema, ed è anche il più facile da dimenticare, perché arriva molto dopo l'analisi.

Quando. Quando l'esito della decisione è noto — settimane o mesi dopo.

Cosa registrare. Due campi:

- `outcome_label` : uno fra `corretta` , `errata` , `mista` , `ignota` . Finché l'esito non è noto, resta `ignota` ; le decisioni `ignota` non concorrono alla calibrazione.
- `decision_changed` : un valore booleano che indica se il contraddittorio ha effettivamente fatto **modificare** la decisione prima di attuarla. È il campo che alimenta il `hit_rate` (Parte IV): senza decisioni cambiate, il `hit_rate` non è calcolabile.

Come scegliere l'etichetta. La qualità della calibrazione dipende dalla coerenza di queste etichette, quindi conviene fissare i criteri *prima* e applicarli sempre uguali. Una convenzione minima che funziona: **corretta** se l'obiettivo dichiarato nella formulazione della decisione è stato raggiunto entro l'orizzonte previsto; **errata** se non è stato raggiunto o si è verificato lo scenario che la decisione escludeva; **mista** se l'esito è parziale o l'obiettivo è stato raggiunto a costi materialmente diversi da quelli previsti. Etichettate sull'esito rispetto all'obiettivo dichiarato, non col senno di poi su ciò che «si sarebbe potuto fare»; se più persone etichettano, concordate i criteri per iscritto — due operatori che etichettano lo stesso esito in modo diverso inquinano la serie più di un esito mancante.

Disciplina. La qualità della calibrazione dipende interamente dalla costanza di questa registrazione. Un tenant che non chiude i propri esiti non potrà mai passare dalla fase descrittiva a una lettura della relazione fra score ed esiti. Vale la pena prevedere un promemoria a distanza per ogni decisione analizzata.

V.4 — CONFIGURARE UN PACK

Un Pack serve quando un dominio ha priorità diverse da quelle generaliste. La configurazione **non richiede modifiche al codice**: è un file di configurazione (Parte II, capitolo II.7).

Passo 1 — Identificare la priorità di dominio. Esempio: in una due-diligence legale la documentabilità delle fonti è critica; conviene alzare il peso di `sources`.

Passo 2 — Ridistribuire i pesi a somma cento. Ogni punto aggiunto a un ingrediente va tolto da un altro. La funzione `validate_weights` rifiuta qualunque configurazione la cui somma non sia 100. Non è possibile spostare le soglie delle tre fasce né alterare la logica del calcolo: quelle restano identiche in ogni dominio.

Passo 3 — Sapere che si è aperta una nuova coorte. Cambiare i pesi rende gli score prodotti dal nuovo Pack non confrontabili con quelli prodotti dalla configurazione precedente. La calibrazione lo gestisce da sé: i pesi in vigore sono registrati su ogni record e fanno parte della chiave di coorte `(version, divergence_method, pesi)`, quindi le due serie non verranno mai mescolate. Resta a voi la disciplina interpretativa: la nuova serie riparte da zero anche ai fini della soglia `MIN_SAMPLE`.

V.5 — LEGGERE L'OUTPUT DI CALIBRAZIONE

La calibrazione si ottiene interrogando l'endpoint `GET /v1/calibration`, che raggruppa i record del ledger per coorte `(version, divergence_method, pesi)` e calcola le statistiche descrittive separatamente per ciascuna (Parte IV). Se non avete accesso diretto all'endpoint, chiedete il report a chi amministra l'installazione.

Cosa leggere per primo — il campo `sufficient`.

- Se `sufficient = False`, la coorte ha meno di cinque esiti definitivi: **qualunque numero prodotto è puramente descrittivo e non va comunicato come validazione**. È il limite «sotto cinque esiti» (Parte III). Fermatevi qui: l'unica azione utile è accumulare altri esiti.

- Se `sufficient = True`, potete leggere la distribuzione per fascia e — se valorizzato — il `hit_rate`.

Come leggere il `hit_rate`. Ricordate la definizione: decisioni cambiate con esito corretto sul totale delle decisioni cambiate con esito definitivo. Può essere `None` anche sopra soglia, se nessuna decisione è mai stata cambiata: non è un'anomalia. E, soprattutto: è **evidenza locale al vostro tenant**, non una verità trasferibile. Un `hit_rate` alto sul vostro dominio non dice nulla su un altro dominio.

Cosa non fare. Non mescolate coorti. Non confrontate un `hit_rate` prodotto con Jaccard con uno prodotto con coseno semantico, né score di versioni o Pack diversi. Il sistema tiene le coorti separate proprio perché quel confronto sarebbe privo di significato.

APPENDICE A

Glossario

CSI (Cognitive Sovereignty Index)	Metrica che traduce il CoglA in qualcosa di misurabile e monitorabile nel tempo, restituendo la postura cognitiva dell'organizzazione e la distribuzione dell'autorità epistemica fra umani e macchine.
Δ-CSI	Metrica operativa dell'engine CounterBrain che misura l'intensità del contraddittorio in una singola decisione; scala 0–100; da non confondere col CSI organizzativo.
Intensità della sfida	Il costrutto misurato dal Δ -CSI: quanta pressione il contraddittorio ha esercitato sulle assunzioni della decisione. Distinta dalla correttezza della decisione, dalla probabilità di successo, dal merito del decisore e dall'effettivo cambiamento della decisione (tracciato dal campo <code>decision_changed</code>).
Contraddittore (non oracolo)	Principio di prodotto per cui il sistema stress-testa la decisione invece di emettere un verdetto. Rendere il sistema «compiacente» è considerato un difetto, non un miglioramento.
Contro-scenario (scenario controintuitivo)	Lo scenario opposto plausibile costruito dal contraddittorio: il rovescio strutturato della tesi ufficiale, contro cui la tesi viene misurata.
Divergenza	Quanto il contro-scenario si allontana dalla tesi. Calcolata in modo lessicale (Jaccard) o semantico (coseno-embedding); i due metodi misurano proprietà diverse e producono valori non confrontabili.
Assunzione implicita	Ciò che la decisione dà per scontato senza dichiararlo. Unico ingrediente pesato anche per gravità (<code>severity</code>).
Test di falsificazione	Verifica concreta ed eseguibile, con criterio d'esito, che potrebbe smentire la tesi.
Domanda d'onere della prova	Domanda che stabilisce chi deve dimostrare cosa prima di procedere.
Fonte / provenienza	Materiale citato a sostegno del contraddittorio, con provenienza dichiarata fra un insieme chiuso di valori (<code>web_search</code> , <code>internal_ledger</code> , <code>internal_doc</code> , <code>osint_module</code> , <code>user_provided</code>).
CAP	La costante (= 5) oltre la quale un ingrediente di conteggio smette di contribuire: regola anti-gonfiaggio.
severity_avg	Media aritmetica [1–5] delle gravità delle assunzioni; modula il componente <code>assumptions</code> in <code>csi-v1</code> . Vale 5 di default quando nessuna severità è dichiarata (retrocompatibilità con <code>csi-v0</code>).

Contradiction Floor	Il controllo (<code>_floor_ok</code>) che verifica che una sfida non sia vuota; se non regge, scatta un retry affilato e, in caso di ulteriore fallimento, un errore controllato.
Fascia	Uno dei tre intervalli di lettura del Δ -CSI: marginale $[0, 33)$, moderato $[33, 66)$, sostanziale $[66, 100]$. Soglie fisse in ogni configurazione.
Coorte	Gruppo omogeneo di score che condividono versione del calcolo, metodo di divergenza e distribuzione dei pesi (<code>version</code> , <code>divergence_method</code> , <code>pesi</code>); la calibrazione non mescola mai coorti diverse.
Tenant	L'organizzazione titolare di un'installazione: i suoi dati (decisioni, sfide, esiti) sono isolati da quelli di ogni altra organizzazione, e ogni evidenza di calibrazione vale solo entro quel perimetro.
Flywheel (volano dell'apprendimento)	Il ciclo a tre fasi decisione → esito registrato nel Ledger → calibrazione: ogni giro accumula evidenza locale al tenant. «Flywheel» e «volano dell'apprendimento» sono lo stesso concetto.
Pack	File di configurazione che adatta il sistema a un dominio (per esempio ridistribuendo i pesi degli ingredienti a somma cento), senza rami di codice per singolo cliente.
Sovereign Decision Ledger	Il registro per-tenant di decisioni, sfide ed esiti; è la base dati che alimenta la calibrazione e costituisce il vantaggio del dato proprietario.
hit_rate	Rapporto fra decisioni cambiate con esito corretto e totale delle decisioni cambiate con esito definitivo; evidenza descrittiva e locale al tenant.
MIN_SAMPLE	La soglia (= 5) di esiti definitivi per coorte sotto la quale la calibrazione resta puramente descrittiva (<code>sufficient = False</code>).
csi-v0 / csi-v1	Le due versioni del calcolo; differiscono solo per il fattore di severità nel componente <code>assumptions</code> . I loro record non sono confrontabili.
System 0	Livello di elaborazione algoritmica che precede e alimenta la cognizione umana. Si colloca sotto il pensiero intuitivo (Sistema 1) e quello analitico (Sistema 2) della teoria del doppio processo, strutturando dati, opzioni e formulazioni prima che il ragionamento cominci.
CogIA (Cognitive Impact Assessment)	Strumento di valutazione dell'impatto dei sistemi di IA sui processi cognitivi dell'organizzazione. Misura tre dimensioni: dipendenza cognitiva, framing effect e capacità di verifica critica.
Thinking Quadrants	Modello delle quattro posture decisionali — esplorativa, generativa, critica e operativa — con cui un decisore affronta l'interazione con l'IA. Ne deriva il valore diagnostico di fenomeni come il bypass diagonale e il posture drift.
Cognitive lock-out	Spostamento progressivo e difficilmente reversibile della strutturazione del pensiero verso sistemi di IA esterni e opachi, con erosione della capacità di deliberazione autonoma. È un esito di lungo periodo, non un singolo errore.

**FIMI (Foreign Information
Manipulation and Interference)**

Manipolazione e interferenza informativa di matrice estera: contenuti falsi o manipolatori immessi da attori ostili per orientare opinioni e comportamenti collettivi. È monitorata a livello europeo dall'EEAS.

La voce Δ -CSI e le voci operative (intensità della sfida, contro-scenario, coorte, hit_rate, Contradiction Floor, Pack, e simili) sono specifiche di questo manuale; le voci CSI, System O, CogIA, Thinking Quadrants, cognitive lock-out e FIMI definiscono i termini del quadro concettuale della sicurezza cognitiva richiamati nel testo.

APPENDICE B

Tre esempi numerici lavorati

Questa appendice mostra il Δ-CSI in azione su **tre casi**, uno per fascia. Il caso centrale (l'acquisizione) è l'esempio golden riproducibile; i due laterali (finanza e sanità) sono **illustrativi**: i componenti sono scelti per mostrare una fascia, ma il calcolo applicato è quello reale e verificabile a mano.

Ogni contributo si ottiene applicando le formule dell'Appendice C. Ricordiamo la regola: i componenti di conteggio usano $\text{_norm}(n) = \min(n, 5) / 5$; **assumptions** è inoltre moltiplicato per $\text{severity_avg} / 5$; **divergence** è già in $[0,1]$ e va moltiplicato per 20.

B.1 — MARGINALE: UNA DECISIONE FINANZIARIA POCO FALSIFICABILE (Δ-CSI 22,00)

Contesto: «mantenere l'attuale allocazione di portafoglio per il prossimo trimestre». Decisione formulata in modo generico, che offre poca presa al contraddittorio. Componenti in ingresso: 2 assunzioni con $\text{severity_avg} = 2,5$; 1 falsificazione; 1 domanda d'onere; nessuna fonte; divergenza euristica 0,40.

COMPONENTE	N / VALORE	CALCOLO	CONTRIBUTO
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	$20 \times 0,40$	8,00
Totale			22,00

22,00 cade nella fascia **marginale** (< 33). Lettura corretta: non «decisione buona», ma «il contraddittorio ha trovato poco» — probabile sintomo di una formulazione troppo vaga, da riscrivere in termini più impegnativi (Parte I, capitolo I.6).

B.2 — MODERATO: L'ACQUISIZIONE (ESEMPIO GOLDEN) (Δ-CSI 54,23)

Contesto: acquisizione di una startup tecnologica. Decisione analizzata: «Procedere con l'acquisizione al multiplo concordato». È lo stesso caso raccontato in forma narrativa nel capitolo I.1. Componenti in ingresso: 3 assunzioni (nessuna severità dichiarata →

`severity_avg = 5,0`, che in `csi-v1` produce lo stesso risultato di `csi-v0`); 2 falsificazioni; 2 domande d'onere; 1 fonte (`user_provided`); divergenza euristica Jaccard 0,9615.

COMPONENTE	N / VALORE	CALCOLO	CONTRIBUTO
<code>assumptions</code>	3, <code>severity_avg</code> 5,0	$25 \times (3/5) \times (5/5)$	15,00
<code>falsifications</code>	2	$25 \times (2/5)$	10,00
<code>burden_questions</code>	2	$20 \times (2/5)$	8,00
<code>sources</code>	1	$10 \times (1/5)$	2,00
<code>divergence</code>	0,9615	$20 \times 0,9615$	19,23
Totale			54,23

Output testuale dello 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 cade nella fascia **moderato** [33, 66): il contraddittorio ha esercitato una pressione moderata sulle assunzioni della decisione.

Questi numeri sono riproducibili in modo deterministico: dati gli stessi cinque componenti, lo scorer restituisce sempre lo stesso numero. Il valore di divergenza `0,9615` è la distanza di Jaccard fra la tesi e lo scenario controintuitivo. Lo snippet seguente replica il calcolo:

```

from brain_engine.csi import CsiComponents, score_delta_csi

# I cinque componenti dell'esempio golden.
# severity non dichiarata → severity_avg = 5 (default storico,
# retrocompatibile con csi-v0).
# divergence = distanza di Jaccard fra tesi e contro-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)

```

Output atteso:

```

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

```

La divergenza può a sua volta essere ricalcolata dai due test con `from brain_engine.csi import heuristic_divergence` e `heuristic_divergence(tesi, contro_scenario)`, che restituisce `0.9615384615384616`.

B.3 — SOSTANZIALE: UN PROTOCOLLO CLINICO MESSO A DURA PROVA (Δ-CSI 78,00)

Contesto: «adottare il nuovo protocollo terapeutico come standard di reparto». Decisione ad alto impatto, contraddetta a fondo. Componenti in ingresso: 5 assunzioni con `severity_avg = 4,6`; 4 falsificazioni; 3 domande d'onere; 3 fonti; divergenza 0,85.

COMPONENTE	N / VALORE	CALCOLO	CONTRIBUTO
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

COMPONENTE	N / VALORE	CALCOLO	CONTRIBUTO
divergence	0,85	$20 \times 0,85$	17,00
Totale			78,00

78,00 cade nella fascia **sostanziale** (≥ 66): il contraddittorio ha messo in discussione in profondità le assunzioni della decisione. Non è un ordine di cambiare protocollo — è l'obbligo di esaminare, una per una, cinque assunzioni gravi, quattro test eseguibili e uno scenario opposto molto distante prima di adottarlo.

NOTA SULLA RIPRODUCIBILITÀ DEI CASI ILLUSTRATIVI

Nei casi B.1 e B.3 la `severity_avg` (2,5 e 4,6) è diversa dal default 5; per riprodurli con lo scorer occorre impostarla esplicitamente, altrimenti il componente `assumptions` userebbe `severity_avg = 5` e i totali salirebbero rispettivamente a 27,00 e 80,00. Le tabelle qui sopra riportano il calcolo esatto con le severità dichiarate.

Le tre fasce a confronto: 22 (marginale, finanza) · 54,23 (moderato, acquisizione) · 78 (sostanziale, sanità). Lo stesso strumento, tre intensità di collaudo diverse — nessuna delle quali dice se la decisione fosse giusta.

APPENDICE C

Riferimento formule completo

Riferimento sintetico e autosufficiente di tutte le regole di calcolo (`csi-v1`).

Normalizzazione dei conteggi: `_norm(n) = min(n, CAP) / CAP`, con `CAP = 5`. Quindi `_norm(0)=0`, `_norm(1)=0,2`, ..., `_norm(5)=1,0`, `_norm( $\geq 5$ )=1,0`.

I cinque contributi:

COMPONENTE	PESO	FORMULA
<code>assumptions</code>	25	<code>25 × _norm(n) × (severity_avg / 5)</code>
<code>falsifications</code>	25	<code>25 × _norm(n)</code>
<code>burden_questions</code>	20	<code>20 × _norm(n)</code>
<code>sources</code>	10	<code>10 × _norm(n)</code>
<code>divergence</code>	20	<code>20 × divergence</code>

Δ -CSI = somma dei cinque contributi. Massimo teorico: 100 (pesi a somma cento).

Severità: `severity_avg` = media [1-5] delle severità delle assunzioni. Singola severità mancante $\rightarrow$ conta 3. Nessuna severità dichiarata nel challenge $\rightarrow$ `severity_avg = 5` (default storico).

Divergenza:

- Jaccard (lessicale): `divergence = 1 - |A  $\cap$  B| / |A  $\cup$  B|` sui token normalizzati.
- Coseno (semantico): `divergence = (1 - coseno) / 2`, coseno mappato da [-1, 1] a [0, 1].
- Selezione: `resolve_divergence` legge `BRAIN_DIVERGENCE` (`auto` | `cosine` | `heuristic`) e la disponibilità di un `EmbeddingProvider` semantico. Fallback garantito: Jaccard.

Fasce (soglie fisse): marginale [0, 33) · moderato [33, 66) · sostanziale [66, 100].

Pesi personalizzati: un Pack può sovrascrivere `csi_weights` se `validate_weights` conferma somma = 100. Soglie e logica restano invariate.

Calibrazione: per coorte (`version`, `divergence_method`, `pesi`); record senza `csi_weights` $\rightarrow$ pesi core; `sufficient = True` solo con $\geq$ `MIN_SAMPLE = 5` esiti definitivi; `hit_rate` = cambiate-corrette / cambiate-definitive (può essere `None`).

APPENDICE D

Casistiche limite

Comportamenti agli estremi, utili per test di regressione e per interpretare score anomali.

- **Nessuna fonte** (`sources: []`). Contributo `sources` = 0. La sintesi è valida e viene scorata; nessun allarme esplicito segnala l'assenza (limite dichiarato, Parte III). Da controllare manualmente.
- **Nessuna severità dichiarata.** `severity_avg` = 5 → `assumptions` si comporta come in `csi-v0`. È il caso dell'esempio golden.
- **Tutte le severità mancanti su singole assunzioni.** Ogni severità assente conta 3 → `severity_avg` = 3 → fattore $3/5 = 0,6$. Diverso dal caso «nessuna severità dichiarata» (che dà 5): la distinzione è deliberata.
- **Conteggio oltre il CAP (`n > 5`).** `_norm` è già saturo a 1,0: elementi aggiuntivi non cambiano il contributo. Uno score non distingue fra 5 e 50 elementi su un componente di conteggio.
- **Divergenza al confine di fascia.** Poiché `divergence` contribuisce fino a 20 punti, un piccolo scarto vicino a una soglia (33 o 66) può cambiare fascia. La fascia va letta come indicazione, non come verità al centesimo.
- **Stessi componenti, versioni diverse.** Con severità dichiarate basse, `csi-v0` e `csi-v1` danno numeri diversi; senza severità dichiarate, coincidono. Mai confrontarli su campioni misti.
- **Encoder irraggiungibile.** La strategia coseno ripiega su Jaccard senza crash; lo score registra come `divergence_method` il metodo effettivamente eseguito (`heuristic`), non quello nominale, ed entra nella coorte corretta (Parte II). Nei record storici prodotti prima di questa correzione il campo può invece riportare il valore nominale.
- **Sfida vuota.** Se `_floor_ok` non regge neppure dopo il retry affilato, l'engine solleva `EngineError`: nessuno score compiacente viene consegnato.
- **`hit_rate` = None sopra soglia.** Campione ≥ 5 ma nessuna decisione cambiata → denominatore zero → `hit_rate` non calcolabile. Non è un errore.
- **Esiti ignota.** Non concorrono al conteggio degli esiti definitivi né al `hit_rate`; una coorte fatta solo di `ignota` resta `sufficient = False`.

APPENDICE E

Changelog delle versioni

Storia delle modifiche al calcolo del Δ-CSE, con l'effetto di ciascuna.

csi-v1 — modulazione per severità delle assunzioni. Il componente `assumptions` passa da conteggio puro ($25 \times \text{_norm}(n)$) a conteggio modulato dalla gravità ($25 \times \text{_norm}(n) \times (\text{severity_avg} / 5)$). Effetto: due challenge con lo stesso numero di assunzioni ma gravità opposta producono ora Δ -CSI distinti. È l'unica differenza formale rispetto a `csi-v0`; tutti gli altri componenti restano identici. La versione è stampata su ogni score nel campo `csi_delta.version` (`CSI_VERSION = "csi-v1"`).

Divergenza semantica come default disponibile. `resolve_divergence` seleziona automaticamente la strategia coseno-embedding quando è disponibile un `EmbeddingProvider` semantico, con fallback euristico Jaccard e `divergence_method` registrato su ogni score. Effetto: la divergenza può misurare distanza di significato, non solo di parole, dove l'ambiente lo consente.

Contraddittorio multi-modello. Le tre passate possono girare su provider diversi (routing controllato da `BRAIN_PASS_PROVIDERS`, con ripiego su un singolo modello), per de-correlare i punti ciechi di un singolo modello. I provider effettivamente usati sono registrati in `meta.models`.

Metodo di divergenza effettivo e coorti per pesi (2026-07-05). Due correzioni di coerenza, senza effetto sulla formula: (1) in caso di ripiego runtime dell'encoder, `divergence_method` registra il metodo effettivamente eseguito, non quello nominale — i record storici possono riportare il valore nominale; (2) la chiave di coorte della calibrazione passa da `(version, divergence_method)` a `(version, divergence_method, pesi)`, usando i `csi_weights` registrati su ogni record (record senza pesi → configurazione core). Contestualmente la stringa `interpretation` diventa «Il contraddittorio ha esercitato una pressione {marginale|moderata|sostanziale} sulle assunzioni della decisione», per aderenza al costrutto misurato.

csi-v0 — versione iniziale. Componente `assumptions` come conteggio puro, senza modulazione per severità. I record `csi-v0` restano leggibili ma non sono confrontabili con quelli `csi-v1` su campioni misti: la calibrazione li tiene in coorti separate.