

SoF *Standard* Definitions.

The definitions behind the metrics operators and investors rely on.

V1.0 · SEP 18, 2026 · CITE AS S0F-V1.0

A metric is a number plus the definition that produced it. Each term below states that definition: the standard, what it measures, how it is computed, a worked example, the choices that change the number, how it is misread, and what it cannot tell you. The canonical, versioned home of every term is <https://www.strategyoffinance.com/standards/>.

Foundations

- 01 Customer — the measured unit
- 02 Revenue types
- 03 Monthly recurring revenue (MRR)
- 04 Annual recurring revenue (ARR)

Movement and retention

- 05 The movement taxonomy
- 06 Pause, churn, and the reactivation window
- 07 Net revenue retention (NRR)
- 08 Gross revenue retention (GRR)
- 09 Cohort retention
- 10 Churn analytics

Structure and quality

- 11 Concentration
- 12 Growth accounting
- 13 ACV and ARPA
- 14 Foreign exchange
- 15 Revenue tie-out
- 16 Definitional sensitivity
- 17 The register

Customer — the measured unit

A customer is the commercial relationship, resolved from its billing accounts, and only paying customers are counted.

ALSO Customer definition · Entity resolution · Resolved parent entity

THE STANDARD

Every customer metric is measured on the *resolved parent entity* — the commercial relationship — not on the billing identifier that happens to appear in the revenue file. Comped and 100%-free customers are excluded from revenue and logo metrics alike.

FOR IMPLEMENTERS

SWITCH

exclude_comped

DEFAULT

true

LOCKED

What it measures

Before any metric can be computed, one question must be answered: what is a customer? The revenue file answers a different question — it lists *billing accounts*, which is not the same thing and is rarely the thing anyone means.

A single commercial relationship routinely appears in a billing system as several identifiers. A customer signs a second contract for a second product and gets a second account id. A subsidiary is invoiced separately from its parent. A company is renamed after an acquisition and the old identifier is retired mid-life. An enterprise migrates from a legacy plan onto a new one and the migration is implemented as “cancel account A, create account B.”

Every one of these is invisible to arithmetic and catastrophic to interpretation. The migration case is the clearest: read at the billing-identifier level, that customer churned and a brand-new customer arrived in the same month. Logo churn rises, new business rises, retention falls, and none of it happened.

How it is computed

Entity resolution runs *before* the customer-month series is built, not after. A versioned resolution maps every raw `customer_ref` in the source data onto a resolved parent entity with a stable id and display name.

All downstream measurement — MRR, movements, cohorts, retention, concentration — operates on the parent.

The resolution is a first-class, versioned object, not a preprocessing step buried in a script. It can be revised; revising it restates every metric that depends on it, which is the correct behavior and the reason it is versioned.

Two attributes attach at the entity level and change what the entity counts toward:

- **Comped** — the customer pays nothing, or pays a nominal amount that does not represent commercial value. Under the locked default they are excluded from revenue *and* logo metrics. They remain visible in the data and in every supporting detail; they are excluded from totals, never deleted.
- **Related party** — the customer is affiliated with the seller. Related-party revenue is not excluded by default, because whether it is arm's-length is a judgment, not a computation. It is flagged so the judgment can be made.

Worked example

A file contains four billing identifiers:

IDENTIFIER	NAME IN FILE	MRR
ACME-001	Acme Corp	7,000
ACME-EU	Acme Corporation (EMEA)	3,000
ACME-LEG	Acme Corp (legacy plan)	0
FRIEND-1	Friendly Design Partner	2,000 (comped)

Unresolved, this reads as four customers, one of which churned to zero, at an average of 3,000 each. Resolved, it reads as **one paying customer at 10,000**, a migration that never churned, and one comped relationship excluded from totals. Concentration, ARPA, logo counts, and retention all change — and only the resolved reading describes the business.

The choices that change the number

- **Where to draw the parent boundary.** A subsidiary billed separately may be the same commercial relationship as its parent or a genuinely separate one. This is a judgment about the business; the standard requires only that it be made explicitly and versioned, not that it come out a particular way.

- `exclude_comped` (LOCKED `true`). Including comped customers inflates logo counts and depresses ARPA, and — more subtly — makes retention look better, because a customer paying nothing cannot contract.
- **Related-party treatment.** Not excluded by default. If the buyer will exclude it, the seller should know the number both ways before the meeting.

How it is misread

The most common error is not making a decision at all — computing at the billing-identifier level because that is what the export contained, and reporting the result as customer metrics. This systematically overstates logo churn, overstates new business, understates ARPA, and understates concentration. It flatters growth and flatters risk simultaneously, which is why it survives so long unchallenged.

The second error is resolving informally — by name-matching in a spreadsheet, unversioned. The metrics then depend on a transformation nobody can reproduce, which means the metrics cannot be reproduced.

What it cannot tell you

Entity resolution cannot tell you whether two identifiers *should* be one relationship. It records a decision; it does not make it. Fuzzy name matching proposes candidates, and a person accepts or rejects them — because the cost of a wrong merge (two real customers collapsed into one, hiding a churn) and the cost of a wrong split are both borne by the reader of the metric, not by the matcher.

Revenue types

Only committed subscription revenue is recurring. Usage, services and one-time fees are real revenue, but never MRR.

ALSO Revenue classification · Recurring vs non-recurring revenue

THE STANDARD

Every revenue line carries exactly one type from a closed seven-value enumeration. Only `recurring_subscription` contributes to MRR. A line whose type cannot be determined is not guessed — the computation stops, or the type is resolved by a person and the resolution is logged.

FOR IMPLEMENTERS

Enumeration (LOCKED): `recurring_subscription` · `usage` · `services` · `one_time` · `transactional` · `credit_refund` · `other_nonrecurring`

What it measures

This is the most consequential classification in the entire standard, and the one most often performed implicitly. “What is your ARR?” is really the question “what portion of your revenue do you believe will still be there next year without another sale?” — and every dispute about that question is a dispute about revenue type.

The enumeration is closed on purpose. An open taxonomy invites a category like “other recurring,” which is where contested revenue goes to avoid scrutiny.

How it is computed

Each type has a specific meaning:

- `recurring_subscription` — a committed, repeating charge for continued access. The subscription persists until someone acts to end it. This is the only type that becomes MRR.
- `usage` — consumption-based charges that vary with volume. Real, often sticky, frequently the fastest-growing line — but not committed. A customer can reduce

usage to zero without cancelling anything.

- **services** — implementation, configuration, training. Typically non-repeating for a given customer and frequently sold at or below cost.
- **one_time** — setup fees, hardware, licence purchases.
- **transactional** — per-event charges that are neither subscription nor metered consumption of the core product.
- **credit_refund** — negative amounts issued against prior billings. Kept as its own type rather than netted silently, because *why* revenue reversed is diagnostic.
- **other_nonrecurring** — the explicit residual. Naming it “other” and marking it non-recurring is a deliberate act: revenue with no better home does not get the benefit of the doubt.

When a source file provides no revenue-type column, the standard does not assume. If the file is genuinely single-type, one fallback is applied to every row and recorded once as an assumption. If the file shows evidence of hybrid revenue — several products, mixed magnitudes, per-product rules — the computation **stops** and asks. A question is cheaper than a wrong ARR.

Where a product column exists but a type column does not, the type may be assigned *per product* (“Email → recurring_subscription, SMS → usage”). The rule applies only to rows with no explicit type of their own, and is recorded as a confirmed assumption naming each product rule.

Worked example

A customer bills 10,000 in a month:

LINE	AMOUNT	TYPE
Platform subscription	6,000	recurring_subscription
API overage	2,500	usage
Onboarding (one-off)	1,000	services
Service credit	(500)	credit_refund

MRR for the month is **6,000**, not 10,000. Annualized run-rate ARR is 72,000. The other 4,000 is real revenue and belongs in every revenue total — it simply does not carry the promise that ARR asserts.

Read carelessly, this customer has “120,000 of ARR.” The gap between 72,000 and 120,000 is not a rounding difference; it is the entire question.

The choices that change the number

- **Usage as recurring.** The single largest lever in the standard. Usage that is contractually committed with a minimum resembles subscription; pure pay-as-you-go does not. This standard keeps them in separate types and lets retention decide whether to *include* usage in its basis — a choice made once, visibly, rather than smuggled into the classification.
- **Multi-year services amortization.** Long implementation programmes can look repeating. They remain **services**; repetition is not commitment.
- **Credits netted vs typed.** Netting credits into subscription revenue hides the reversal rate.

How it is misread

The characteristic failure is *aspirational classification*: revenue that management expects to repeat is typed as recurring before it has repeated. This is rarely dishonest. It is usually a genuine belief, arrived at without the discipline of asking what would happen if the customer simply stopped buying.

A second failure is silent single-typing — a file with no type column read as 100% recurring because that was the default. This produces an ARR that is simply total revenue $\times 12$, a number that will not survive its first serious question.

What it cannot tell you

Classification cannot tell you whether recurring revenue is *durable*. A subscription cancellable at thirty days' notice and one with three years remaining are both **recurring_subscription**. Durability lives in contract terms — see committed vs run-rate ARR — not in the type.

Monthly recurring revenue

MRR is what a customer's subscription pays per month, taken at month end, if nothing changes.

ALSO MRR · Recurring run rate

THE STANDARD

MRR is the *month-end recurring run rate* of a resolved customer: subscription revenue normalized to a monthly rate, held through qualifying pauses, stated in base currency. It is a state, not a flow — and it is deliberately not the same number as recognized revenue.

FOR IMPLEMENTERS

SWITCH

DEFAULT

reactivation_window_months

3

exclude_comped

true

LOCKED

What it measures

MRR answers: *if nothing changes, what does this customer pay us next month?* That question is about a **state at a point in time**, which is why MRR is not simply “revenue divided by months.” Every downstream measure — movements, the bridge, retention, cohorts, concentration — is built on this state series, so an error here propagates everywhere.

The standard carries three parallel views of recurring revenue per customer-month, because a single number cannot answer every legitimate question:

VIEW	WHAT IT HOLDS	WHAT IT ANSWERS
mrr	Month-end run rate; pauses held	What will recur?

VIEW	WHAT IT HOLDS	WHAT IT ANSWERS
<code>recurring_recognized</code>	Rev-rec view: prepaid spread over term, prorated months as billed, nothing during unpaid pauses	What ties to the P&L?
<code>recurring_billed</code>	Raw billed amounts at their billed period, lump sums intact	What did we invoice?

Confusing these three is the most common source of “your ARR doesn’t tie” in diligence. They are all correct; they answer different questions, and the gaps between them are themselves informative — see [tie-out](#).

How it is computed

- 01 **Filter to type.** Only `recurring_subscription` lines contribute. Usage, services, one-time, transactional and credits are carried on the customer-month row as separate measures and excluded from MRR.
- 02 **Normalize the period grain.** A row covering a quarter or a year is recognized evenly across the months it covers. Grain is *declared by a person* when the file is described, never inferred silently from row spacing.
- 03 **Normalize the amount basis.** A file exporting *annualized run-rate* (“ARR by customer”) carries figures twelve times the monthly rate. Declaring the basis converts it. Reading an ARR export as monthly revenue overstates MRR by 12× — the single most destructive input error in this domain, and the reason the basis is an explicit, confirmed declaration.
- 04 **Spread prepaid lump sums** across the contract term where a contract is known, rather than spiking one month.
- 05 **Follow contracted ramps.** Where a contract specifies a ramp schedule, the run rate follows the contracted step, and the step-up is recognized as expansion.
- 06 **Take the month-end state for mid-month changes.** A customer who upgrades on the 15th is billed a prorated amount but *ends* the month at the new rate. MRR takes the month-end rate; the prorated billing shows in the recognized and billed views. Proration is flagged so the difference is traceable.
- 07 **Hold through qualifying pauses.** A billing gap no longer than the reactivation window, followed by resumption, holds MRR at the pre-gap level rather than dropping to zero and back. See [pause vs churn](#).
- 08 **Convert to base currency** at the month’s flow rate. See [FX](#).

Worked example

A customer on an annual contract of 24,000, billed once in January, upgrading to 30,000 annualized effective 15 April:

MONTH	RECURRING_BILLED	RECURRING_RECOGNIZED	MRR
Jan	24,000	2,000	2,000
Feb	0	2,000	2,000
Mar	0	2,000	2,000
Apr	3,000	2,250	2,500
May	0	2,500	2,500

April is the instructive row. Billed shows the upgrade invoice; recognized shows half a month at each rate; MRR shows the month-end run rate of 2,500 — the number that answers “what recurs next month.” Reading the billed column as MRR would show a customer whose revenue collapsed 87% in February.

The choices that change the number

- **Declared grain and basis.** Not really choices — they are facts about the file — but they must be *declared*, and a wrong declaration is catastrophic rather than merely imprecise.
- **reactivation_window_months** (default 3). Governs whether a gap holds MRR or drops it. Shortening the window converts pauses into churn-and-win-back pairs, which lowers retention and raises both gross churn and new business.
- **exclude_comped** (LOCKED **true**). Comped customers carry no MRR toward totals.
- **Contract-informed spreading.** Without contracts, a lump sum can only be recognized where billed; with them, it spreads. This changes the recognized view, never the run-rate view.

How it is misread

The dominant error is treating MRR as a flow — dividing a period’s total revenue by the number of months. That silently includes services and one-time revenue, and it averages away exactly the month-end state the metric exists to capture.

The second is the 12× error described above: an ARR-basis export read as monthly. It is easy to spot once suspected (the ARR figure will be implausibly large relative to reported revenue) and nearly invisible until then, which is why tie-out is never optional.

What it cannot tell you

MRR is a run rate, not a promise. It says what recurs if nothing changes; it says nothing about whether the customer has thirty days' notice or three years remaining. For that, see committed ARR. It also says nothing about collection — MRR counts billed-and-recognized revenue, not cash received.

[HTTPS://WWW.STRATEGYOFFINANCE.COM/STANDARDS/MONTHLY-RECURRING-REVENUE/](https://www.strategyoffinance.com/standards/monthly-recurring-revenue/) · SOF-V1.0

Annual recurring revenue

ARR is this month's MRR times twelve: a run rate, not a forecast and not last year's revenue.

ALSO ARR · Run-rate ARR · Committed ARR · Contracted not yet live

THE STANDARD

ARR is in-month MRR $\times$ 12. It is a *restatement of the current run rate*, not a forecast, not a sum of the trailing twelve months, and not the value of the contract book.

FOR IMPLEMENTERS

SWITCH

DEFAULT

`arr_basis`

`run_rate`

alternative `contract_acv`

What it measures

ARR exists to make a monthly run rate legible at the scale businesses are discussed and valued at. That is its entire job. Every problem with ARR comes from asking it to do more.

Three distinct quantities are routinely all called “ARR”:

- **Run-rate ARR** — current MRR $\times$ 12. What recurs today, annualized. *This is the standard's default.*
- **Committed ARR** — run rate plus the contracted increases already signed (final ramp steps not yet reached), for live contracts.
- **Contracted, not yet live** — signed contracts whose service has not started. Real, valuable, and **not** part of ARR, because ARR describes what is recurring now.

Reporting the second or third as though it were the first is not a rounding difference. It is a different claim about the business.

How it is computed

Run rate (default). ARR at month m = MRR at month $m \times 12$. Nothing else. It is not the sum of the last twelve months of revenue — that quantity is trailing revenue, which for any growing business is materially lower and answers a different question.

Committed. Starting from run-rate MRR, for each live contract carrying a ramp schedule, if the *final* contracted monthly step exceeds the customer's current run rate, the difference is added. Contracts that have ended are excluded; contracts that have not started are excluded from committed and reported separately as contracted-not-yet-live.

Committed ARR requires contract data. Where contracts are absent, the standard reports run rate only and says so rather than inferring commitment from billing patterns.

Worked example

Four customers at a March cutoff:

CUSTOMER	MRR (MAR)	CONTRACT DETAIL
A	5,000	flat, 18 months remaining
B	3,000	ramps to 6,000 in September
C	2,000	contract ends 31 March, not renewing
D	0	signed; service starts 1 May at 4,000/mo

- **Run-rate ARR** = $(5,000 + 3,000 + 2,000) \times 12 = \mathbf{120,000}$. Customer C is included because C is still live in March; the non-renewal shows as churn in April, and because C's contract has expired the trailing-gap rule treats it as definite churn rather than indeterminate.
- **Committed ARR** = run rate + B's ramp gap $(6,000 - 3,000 = 3,000/\text{mo}) = 120,000 + 36,000 = \mathbf{156,000}$.
- **Contracted, not yet live** = D at 4,000/mo = **48,000**, reported as its own line, never inside either ARR figure.

A company describing itself as "at 204,000 ARR" — committed plus contracted-not-yet-live — is not lying so much as answering a question nobody asked. The three figures are 120,000, 156,000 and 48,000, and only the first is ARR under this standard.

The choices that change the number

- **arr_basis** . **run_rate** (default) annualizes current MRR. **contract_acv** derives from contract values instead, which shifts the measure from *what is recurring* to *what is contracted* and materially changes ramp-heavy books.
- **Whether usage is in the base.** ARR inherits MRR's definition, and MRR excludes usage by type. Companies with large metered lines often quote an "ARR" that includes annualized usage; that figure is legitimate only if labelled, because usage can fall without anyone cancelling.
- **The month chosen.** ARR is a spot measure. A month with an unusual billing event annualizes that oddity twelvefold. Seasonal businesses should be read on a trend, never a point.

How it is misread

Peak-month ARR. Selecting the best recent month and annualizing it. Detected by comparing the ARR month against the trend on the ARR/MRR build.

Annualizing non-recurring revenue. A one-time implementation fee multiplied by twelve is the most expensive arithmetic in software finance.

Committed ARR reported as ARR. Defensible as a metric, indefensible as a label. The gap between the two is exactly the contracted-but-not-yet-earned increase, and a reader is entitled to see it separately.

ARR compared against trailing revenue. A company growing 100% year over year will show run-rate ARR far above trailing twelve-month revenue. Neither number is wrong; comparing them without noting the difference produces a phantom discrepancy — or conceals a real one.

What it cannot tell you

ARR is a snapshot of a run rate and cannot express duration, cancellability, concentration or collectability. A book of 120,000 ARR from one customer on thirty days' notice and 120,000 from forty customers on three-year terms are identical in ARR and entirely different businesses. Those differences live in concentration, contract terms, and retention.

The movement taxonomy

Every change in a customer's MRR is exactly one of five movements, and they add up to the closing balance without a plug.

ALSO MRR bridge · Revenue bridge · MRR movements · New, expansion, contraction, churn, reactivation

THE STANDARD

Every month-over-month change in a customer's MRR is classified as exactly one of five movements — new, expansion, contraction, churn, reactivation — signed so that opening MRR plus all movements equals closing MRR, by construction.

FOR IMPLEMENTERS

Movement types (LOCKED): `new` · `expansion` · `contraction` · `churn` · `reactivation`

What it measures

The bridge answers the question that a growth rate cannot: *where did the change come from?* Two companies both growing 30% are different businesses if one grew on new logos while the other grew on expansion within a shrinking customer base.

The taxonomy is mutually exclusive and collectively exhaustive. That is not a stylistic preference — it is what makes the bridge an identity rather than a reconciliation:

```
closing = opening + new + expansion + reactivation + contract
```

Contraction and churn carry negative signs, so the identity is a plain sum. If this does not tie exactly, something has been double-counted or dropped, and the error is in the classification, not in the arithmetic.

How it is computed

Movements are derived from the *held* MRR series — the series after pause handling and currency conversion, so pause artifacts never appear as churn. For each customer and each month, take the delta from the prior month:

PRIOR	CURRENT	CLASSIFICATION
0	> 0	new if never previously active, else reactivation
> 0	0	churn
> 0	higher	expansion
> 0	lower (non-zero)	contraction
any	unchanged	no movement

A zero delta produces no movement at all. Every non-zero delta produces exactly one, which is what guarantees the identity.

Three sub-tags may attach to a movement without changing its type:

- **cross_sell** — expansion coinciding with a product the customer had not previously bought.
- **ramp_step** — expansion arising from a contracted ramp schedule rather than a new sale. Materially different in quality: it was already signed.
- **fx_driven** — the change in base currency arises purely from exchange rates while native amounts were unchanged. Sub-totalled in the bridge and removed entirely in the constant-currency view.

Worked example

A customer's held MRR series and the movements it generates:

MONTH	MRR	DELTA	MOVEMENT
Jan	0	—	—

MONTH	MRR	DELTA	MOVEMENT
Feb	1,000	+1,000	new
Mar	1,500	+500	expansion
Apr	1,200	−300	contraction
May	0	−1,200	churn
...	0	0	— (gap beyond window)
Sep	800	+800	reactivation

Opening (Jan) 0 + 1,000 + 500 − 300 − 1,200 + 800 = **800** = closing (Sep). The identity holds without adjustment, because every delta was classified once.

Note that Sep is **reactivation**, not **new**. The distinction matters: a win-back is not the same commercial event as a first-time acquisition, and conflating them overstates new-business productivity while hiding a retention problem.

The choices that change the number

- **The reactivation window** determines whether a gap produces a churn/reactivation pair or is absorbed as a pause with no movements at all. This is the largest single lever on both churn and new-business figures. See [pause vs churn](#).
- **Netting at the customer level.** This standard classifies the *net* change per customer-month. A customer who adds one product and drops another in the same month produces a single net movement, not an offsetting pair. Reporting gross-up-and-down separately is legitimate but is a different presentation, and mixing the two double-counts.
- **Whether FX-driven changes are movements at all.** Here they are movements with a sub-tag, and the constant-currency view removes them — so the reader can see both the reported bridge and the operating bridge.

How it is misread

New and reactivation reported together as “new.” Flatters acquisition and conceals churn-and-return churn.

Downgrades netted into churn. Contraction and churn have different implications — one is a customer who reduced, the other a customer who left. Collapsing them makes churn look worse and expansion look better

than the truth.

A bridge that doesn't tie, adjusted with a plug. A "reconciling item" line in a bridge is a confession that the taxonomy was not exhaustive. Under this standard a plug cannot arise; if the numbers disagree, the classification is wrong and must be fixed rather than balanced.

What it cannot tell you

The taxonomy describes *what* changed, never *why*. Contraction from a seat reduction, a negotiated discount, and a downgrade after a service failure are identical in the bridge. Reason codes do not exist in customer-level revenue data; where they matter, they must come from CRM or churn-reason systems and be joined deliberately.

[HTTPS://WWW.STRATEGYOFFINANCE.COM/STANDARDS/MOVEMENT-TAXONOMY/](https://www.strategyoffinance.com/standards/movement-taxonomy/) · SOF-V1.0

Pause, churn, and the reactivation window

A short billing gap that closes is a pause. A long one is churn and a win-back. A gap still open at the cutoff is judged separately.

ALSO [Reactivation window](#) · [Win-back](#) · [Billing gap treatment](#) · [Right-censored churn](#)

THE STANDARD

A completed billing gap no longer than the reactivation window is a **pause**: MRR is held at the pre-gap level and no movements are recorded. A completed gap longer than the window is **churn at the gap start, reactivation on return**. A gap still open at the data cutoff is *right-censored* and governed separately.

FOR IMPLEMENTERS

SWITCH

DEFAULT

`reactivation_window_months`

`3`

`trailing_gap_policy`

`churn_at_gap_start`

alternative `hold_as_paused`

What it measures

A customer bills, stops for two months, then bills again. Did they churn and come back, or did they never leave?

Nothing in the data answers this. The revenue file records absence, not intention. Yet the answer moves churn, retention, new business and cohort curves simultaneously — which makes it exactly the kind of choice that must be made once, explicitly, and disclosed, rather than made implicitly by whichever analyst wrote the query.

How it is computed

For each customer, the MRR series is scanned for runs of zero months that occur *after* the customer has been active at least once. Leading zeros — before the first activity — are not gaps; the customer had not yet arrived.

Each zero-run is then resolved by two questions: **does it end?** and **how long is it?**

Completed gap, length $\leq$ window $\rightarrow$ pause. MRR is held at the pre-gap level for every month of the gap. The customer never drops to zero, so no churn and no reactivation appear. An assumption is logged naming the customer, the gap start, the length, and the held amount; an informational exception records the gap.

Completed gap, length $>$ window $\rightarrow$ churn and reactivation. The series is left at zero, which produces **churn** at the gap start and **reactivation** on return through the ordinary movement rules. An exception records both dates.

Open gap at the cutoff $\rightarrow$ right-censored. The rule above speaks to completed gaps; a customer inactive at the cutoff may be gone, or may return next month into data that does not exist yet. Two sub-cases:

- **Contracts all expired before the gap began $\rightarrow$ definite churn.** Expiry without renewal is not ambiguous, and the standard says so rather than hedging.
- **Otherwise, gap shorter than the window $\rightarrow$ genuinely indeterminate.** The configured **trailing_gap_policy** applies (default: churn at gap start, the conservative reading), a **warn**-severity **indeterminate_churn** exception is raised naming the customer, and the alternative is recorded so the toggle restates cleanly.

Worked example

Window = 3 months. Cutoff = December.

CUSTOMER	PATTERN	TREATMENT
A	Bills Jan–Mar, silent Apr–May, bills Jun+	2-month gap $\leq$ 3 $\rightarrow$ pause ; MRR held at March level through Apr–May; no movements
B	Bills Jan–Mar, silent Apr–Aug, bills Sep+	5-month gap $>$ 3 $\rightarrow$ churn in Apr, reactivation in Sep
C	Bills Jan–Oct, silent Nov–Dec, contract runs to next June	Open gap, 2 months $<$ window $\rightarrow$ indeterminate ; policy applied, flagged warn
D	Bills Jan–Oct, silent Nov–Dec, contract ended in October	Open gap, contracts expired $\rightarrow$ definite churn in Nov

A and B differ only in the length of a silence. Their retention contributions, churn counts and cohort curves diverge entirely — from one configuration value.

The choices that change the number

- **reactivation_window_months** (default 3). The dominant lever. Lengthening it converts churn/reactivation pairs into pauses: churn falls, retention rises, new-business-adjacent reactivation falls. Shortening it does the reverse. The default of 3 is a deliberate midpoint: long enough to absorb ordinary billing irregularity (a delayed invoice, a renewal processed late), short enough that a genuinely departed customer is not carried for two quarters.
- **trailing_gap_policy** . **churn_at_gap_start** (default) treats right-censored silence as departure — conservative, and the reading a buyer will apply. **hold_as_paused** treats it as continuing. Every affected customer is individually listed, so the difference can be examined name by name rather than argued in the abstract.

How it is misread

Window chosen to flatter. A twelve-month reactivation window means almost nobody ever churns; retention approaches 100% by construction. The tell is a window materially longer than the billing cycle it is meant to absorb.

Pause treatment applied to right-censored gaps. The most common quiet error: holding MRR for customers who have simply gone silent at the cutoff, which props up closing ARR with customers who may already be gone. The standard separates the completed case from the censored one precisely because the evidence is different.

Reactivations counted as new. Discussed under [movements](#); it originates here.

What it cannot tell you

No window can distinguish a paused customer from a departed one *at the time*. The rule is a disclosure convention, not a discovery. What it guarantees is that the same rule was applied to everyone, that every affected customer is named, and that changing the rule restates every dependent number rather than quietly changing one.

Net revenue retention

NRR asks what last year's customers are worth this year, expansion included and new customers excluded.

ALSO NRR · Net dollar retention · NDR · Net revenue retention rate

THE STANDARD

NRR measures what a fixed set of customers is worth now versus what it was worth then, expansion included and new customers excluded. The SoF default is the **monthly-cohort compounded trailing twelve months**, usage-inclusive and reactivation-inclusive. Two alternative variants are defined, and all three are reported.

FOR IMPLEMENTERS

SWITCH	DEFAULT	
<code>nrr.default_variant</code>	<code>ttm_compounded</code>	alternatives <code>yoy_snapshot</code> , <code>quarterly_annualized</code>
<code>nrr.usage_inclusive</code>	<code>true</code>	LOCKED
<code>nrr.reactivation_inclusive</code>	<code>true</code>	
<code>nrr.usage_smoothing</code>	<code>none</code>	trailing-three-month smoothing available, never default

What it measures

NRR is the closest thing software finance has to a single-number verdict on product-market fit: does the installed base grow on its own? Above 100% means existing customers spend more this year than last, net of everything lost — the business compounds without new sales. Below 100% means new sales are refilling a leaking bucket.

Because so much rides on it, NRR is also the most definitionally fragile number in the standard. The same customer base can produce materially different NRRs — all of them honestly computed — depending on four choices. This standard names all four and computes the alternatives rather than picking one and hoping.

How it is computed

Variant (a) — TTM compounded (SoF default). For each of the trailing twelve months, compute a month-over-month cohort ratio: take the customers active in month $m-1$, sum their basis revenue in $m-1$ (denominator), sum the same customers' revenue in m (numerator). Multiply the twelve monthly ratios together.

Months with no retained base are skipped rather than treated as zero — a zero-base month would annihilate the product.

Compounding is the most faithful of the three because a customer that joins mid-year enters the calculation as soon as it has a prior month, and the measure never depends on a single arbitrary base month.

Variant (b) — YoY same-customer snapshot. Take the customers active twelve months ago. Denominator: their revenue then. Numerator: the same customers' revenue now. One ratio, two points in time. Intuitive and widely quoted; blind to everything that happened in between, and heavily influenced by the choice of base month.

Variant (c) — Quarterly annualized. Compare the trailing three months against the three before, for customers active in the earlier window, then raise the ratio to the fourth power. Responsive to recent movement; amplifies noise and seasonality by construction, since a single soft quarter is compounded four times.

The basis. In all three, "revenue" means MRR, plus usage when `usage_inclusive` is true (the locked default). Comped customers are excluded.

Reactivation. When `reactivation_inclusive` is true (default), a customer who returns within the window rejoins the numerator — they were part of the historical base and their return is a genuine recovery. When false, the treatment is stricter: in the YoY variant, customers with any churn event inside the window are removed from the base entirely, so a departure-and-return cannot be presented as retention.

Worked example

Three customers, measuring December against the prior December:

CUSTOMER	DEC (BASE)	DEC (NOW)	NOTE
A	10,000	12,000	expanded
B	8,000	0	churned in June
C	5,000	5,000	flat
D	0	9,000	new — excluded

YoY snapshot = $(12,000 + 0 + 5,000) \div (10,000 + 8,000 + 5,000) = 17,000 \div 23,000 = 73.9\%$.

Customer D's 9,000 is real revenue and does not appear. That exclusion is the metric: NRR asks what the existing base did, and adding new customers to the numerator would produce a growth rate wearing a retention label — an error severe enough that a "retention" figure above the company's total growth rate should always be checked for it.

Note also that the TTM compounded variant on this same data would differ, because it sees B's decline in the month it happened and compounds around it.

The choices that change the number

- **Variant.** Can move the figure by tens of points on the same data. Not a matter of correctness — a matter of what question is being asked.
- **usage_inclusive** (LOCKED `true`). For a usage-heavy business this is the largest single lever. Including usage credits genuine consumption growth; excluding it isolates committed subscription. Both are legitimate; quoting one without saying which is not.
- **reactivation_inclusive** (default `true`). Whether a win-back repairs retention or is excluded as a break in the relationship.
- **usage_smoothing**. Trailing-three-month smoothing of usage is available and is *never* the default: smoothing suppresses exactly the volatility that makes usage revenue different from subscription revenue.
- **Everything upstream.** The reactivation window, comped exclusion, and entity resolution all move NRR before the retention calculation begins.

How it is misread

Quoting NRR without its variant and basis. The number is uninterpretable without them. This is the single most common failure in the category.

Dollar-weighted NRR read as typical customer behavior. NRR is revenue-weighted: one large customer's expansion can carry an entire portfolio while the median customer contracts. Always read alongside logo retention and the cohort triangles.

Survivor-biased bases. Computing over customers present at *both* endpoints excludes everyone who churned, which produces a figure that cannot go below 100% by construction. The base must be fixed at the start.

Comparing across companies. Two firms' NRRs are comparable only if variant, basis, window and entity resolution match — which, absent a shared standard, is almost never true.

What it cannot tell you

NRR cannot distinguish price increases from volume growth, cannot separate contractual escalators from genuine adoption, and says nothing about the cost of achieving the expansion. A base retained by heavy discounting and one retained by product value are indistinguishable in NRR alone.

Gross revenue retention

GRR is NRR with every customer capped at what it paid at the start, so growth cannot hide leakage.

ALSO GRR • Gross dollar retention • GDR

THE STANDARD

GRR measures the same fixed customer base as NRR, with each customer **capped at its own starting revenue**. Expansion cannot offset loss. GRR is therefore bounded above by 100% and is computed under the same three variants as NRR.

FOR IMPLEMENTERS

Identical to net revenue retention — the only difference is the per-customer cap.

What it measures

NRR answers “did the base grow?” GRR answers the harder question: “**did the base leak?**”

The distinction matters because NRR can conceal severe churn behind concentrated expansion. A business where one large account triples while a third of the customer base departs can post NRR above 100% — a headline that reads as health and describes a business in trouble. GRR removes the concealment by refusing to let any customer contribute more than it started with.

The gap between NRR and GRR is itself the most useful reading: it is the expansion rate of the retained base. A wide gap means growth depends on expanding a shrinking group; a narrow gap means the base is stable but not growing. Neither is inherently better, and both are invisible if only one of the two is quoted.

How it is computed

Identical to NRR in every respect — same base cohort, same window, same variant mechanics, same basis and toggles — with one change to the numerator:

```
numerator =  $\sum \min(\text{revenue\_now}(\text{customer}), \text{revenue\_base}(\text{customer}))$ 
```

Each customer contributes the *lesser* of its current and starting revenue. Growth is truncated; decline is counted in full. Because every term is capped at its base contribution, the ratio cannot exceed 1.0, which is why a GRR above 100% is not an impressive result but a computational error — most often a base set that was not fixed, or a cap applied to the aggregate rather than per-customer.

That last error deserves emphasis: capping the *total* rather than each *customer* lets one account's expansion mask another's contraction and destroys the measure. The cap is per customer, always.

Worked example

The same four customers used in the NRR entry:

CUSTOMER	BASE	NOW	NRR CONTRIBUTION	GRR CONTRIBUTION
A	10,000	12,000	12,000	10,000 (capped)
B	8,000	0	0	0
C	5,000	5,000	5,000	5,000
Total	23,000	17,000	17,000	15,000

— $\text{NRR} = 17,000 \div 23,000 = \mathbf{73.9\%}$

— $\text{GRR} = 15,000 \div 23,000 = \mathbf{65.2\%}$

The 8.7-point gap is precisely A's 2,000 of expansion. Reading both tells you the base lost roughly a third of its revenue and clawed back a fraction through one customer's growth. Reading NRR alone tells you far less.

The choices that change the number

All of NRR's choices apply identically. Two are worth restating because they interact differently with the cap:

- **usage_inclusive**. Usage volatility is asymmetric under a cap: a month of unusually high consumption is truncated away, while a month of unusually low

consumption counts fully. Usage-inclusive GRR therefore reads lower and noisier than intuition suggests, and the effect is structural rather than a sign of deterioration.

- **reactivation_inclusive**. A returning customer capped at its original revenue can restore GRR at most to where it began.

How it is misread

Reported above 100%. Always an error. Investigate the base construction before investigating the business.

Used interchangeably with logo retention. GRR is revenue-weighted. Losing one large customer and losing twenty small ones can produce the same GRR and mean entirely different things. Pair it with the logo cohort triangle.

Quoted without NRR (or vice versa). The pair is the information. Either alone invites a conclusion the other would refuse.

What it cannot tell you

GRR measures revenue leakage, not its cause, and — like NRR — is silent on whether retention was purchased with discounts, credits or concessions. A base retained by repricing downward shows as contraction, not churn; the customer stayed, the revenue did not, and only the bridge distinguishes the two.

Cohort retention

Cohort retention follows each vintage of customers on its own curve, measured at the end of each period against what they paid when they joined.

ALSO Cohort analysis • Retention triangle • Vintage retention • Logo retention

THE STANDARD

A cohort is the set of customers whose **first month with MRR above zero** falls in a given period. Each member's base is its MRR **in its own joining month**; each cell measures the cohort at the **last month of the period being reported**, clamped to the data cutoff. Cohort triangles use an MRR-only basis: usage is excluded by design.

FOR IMPLEMENTERS

SWITCH

`exclude_comped`

DEFAULT

`true`

LOCKED

Granularity (monthly, quarterly, yearly) is a view chosen at read time, not a stored definition.

What it measures

Aggregate retention tells you how the base behaved. Cohort retention tells you whether the *business is changing* — because it separates customers by when they arrived and lets each vintage be read on its own curve.

This is the only view that answers questions such as: are customers acquired this year retaining better than those acquired two years ago? Does the decay flatten after a certain age, and where? Did a pricing or packaging change improve the durability of what came after it?

Two triangles are produced. The **revenue triangle** tracks MRR, in both a net form (expansion included) and a gross form (each member capped at its own starting MRR, as in GRR). The **logo triangle** tracks how many of the cohort's customers are still active. They diverge in an informative way: revenue retention holding up while logo retention falls means the survivors are expanding enough to mask departures — durable-looking revenue on a narrowing base.

How it is computed

- 01 **Assign cohorts.** A customer's cohort is its first month with MRR above zero. Comped customers are excluded under the locked default, and a customer with no qualifying month has no cohort.
- 02 **Fix each member's base at its own joining month.** Not at the cohort period's start. This matters as soon as the granularity is coarser than monthly: a customer joining in March belongs to Q1, and its base is its March MRR.
- 03 **Measure each cell at the period end.** For a quarterly cohort, cell k is measured at the last month of the k -th quarter after the cohort's quarter began; for yearly, the last month of the k -th year. A period still in progress at the data cutoff is measured at the cutoff rather than dropped.
- 04 **Compute the ratios.** Net = Σ members' MRR at the measured month $\div$ Σ bases. Gross = Σ min(MRR at measured month, own base) $\div$ Σ bases. Logos = count active at the measured month $\div$ count active at base.

Period-end anchoring is not a stylistic choice; measuring at the period *start* treats every member who joined later in the period as an immediate total loss. A customer joining in March and never churning would read 0% at its own Q1 cell — an artifact of the anchor, indistinguishable at a glance from catastrophic churn.

Worked example

A 2024-Q1 cohort with two members, both retained:

MEMBER	JOINED	MRR AT JOINING
Alpha	2024-01	100
Beta	2024-03	400

Base = 500. Cell 0 is measured at **2024-03** (the quarter's last month), where both are live: $500 \div 500 = 100\%$. Cell 1 is measured at 2024-06, and so on.

Measured at the quarter's *start* instead, cell 0 would read $100 \div 500 = 20\%$, because Beta did not exist in January. Same customers, same perfect retention, an 80-point difference from the anchor alone.

The choices that change the number

- **Granularity.** Monthly cohorts are the most precise and the noisiest — small monthly cohorts produce violent percentages. Quarterly is usually the best compromise; yearly is for long-horizon shape. All three are computed from the same underlying customer-months, so switching restates nothing.

- **The cohort trigger.** First MRR month is the standard. First *invoice* would place customers a month or two earlier for prepaid contracts; first *contract signature* earlier still. The triangle shifts accordingly.
- **MRR-only basis (by design).** Usage is excluded from cohort triangles even when retention elsewhere is usage-inclusive, because usage volatility swamps the subscription decay curve the triangle exists to show. This is a deliberate divergence from the NRR basis and is stated on the table.
- **Comped exclusion,** as everywhere.

How it is misread

Reading immature cohorts as trends. The most recent cohort has the fewest observable periods and the smallest sample. Its early cells are not evidence of improvement.

Comparing cells at different ages. A cohort at month 6 and one at month 24 are not comparable. Triangles are read *down the columns* — same age, different vintages — not across rows.

Small-cohort noise treated as signal. A cohort of three customers moves 33 points when one leaves. Always read cell values against the cohort's base size.

Averaging the triangle into one number. The triangle's information is its shape. Collapsing it discards exactly what it was built to reveal.

What it cannot tell you

Cohort retention describes what happened to a vintage; it cannot attribute the cause. A cohort acquired during a promotional period may retain worse because of the discount, the channel, the segment, or the macro environment — the triangle shows the divergence and is silent on which. Slicing cohorts by segment or channel narrows the question but never closes it.

Churn analytics

Churn is counted in the first month a customer pays nothing, and reported four ways: by logo, by dollar, by tenure, and by survival.

ALSO Churn rate · Logo churn · Dollar churn · Survival curve · Tenure at churn

THE STANDARD

Churn is recognized in the **first month with zero MRR** after a qualifying gap — MRR is held through the last active month, not truncated at the last invoice date. Churn is reported in four views: monthly logo and dollar rates, tenure at churn, calendar seasonality, and survival by tenure.

FOR IMPLEMENTERS

SWITCH

DEFAULT

reactivation_window_months

3

trailing_gap_policy

churn_at_gap_start

thresholds.churn_seasonality_sigma

2

What it measures

“Churn rate” is four different measurements wearing one name, and the confusion between them is routine.

- **Logo churn** — what share of customers left. Democratic: every customer counts once.
- **Dollar churn** — what share of revenue left. Weighted: one large departure can exceed twenty small ones.
- **Tenure at churn** — how long customers lasted before leaving. Distinguishes an onboarding failure from a renewal failure.
- **Survival** — of customers observable at tenure k , how many were still active. The clearest view of *when* a business loses people.

Logo and dollar churn diverging is diagnostic, not contradictory. Low logo churn with high dollar churn means the business is losing few but large customers — a concentration problem. The reverse means it is shedding a long tail while holding its core.

How it is computed

Timing. MRR is held through the customer's last active month; churn is recognized in the first zero month thereafter, subject to the pause rules in [pause vs churn](#). A customer whose final invoice covers March churns in April, not March. This one-month convention matters when comparing figures across systems: an off-by-one convention shifts every cohort curve and every monthly rate.

Monthly rates. For each month: churned logos and churned MRR (the absolute value of churn movements), against the prior month's active base and base MRR. Reporting both numerator and denominator, rather than only the ratio, lets a reader see when a rate moved because the base changed rather than because losses did.

Tenure at churn. For each churn event, the months between the customer's cohort month and the churn month, with the MRR lost.

Seasonality. Churn events and lost MRR grouped by calendar month across all years, so an annual renewal concentration becomes visible. A month exceeding the mean by more than `churn_seasonality_sigma` standard deviations is flagged. Seasonality is expected in businesses with January or fiscal-year-end renewal cycles; its *absence* in such a business is the anomaly.

Survival. For each tenure k , count the customers whose cohort month plus k falls at or before the data cutoff (**observable**), and how many of those had MRR above zero at that point (**surviving**). Restricting the denominator to observable customers is what makes the curve honest: a customer acquired two months before the cutoff simply cannot inform month-24 survival, and including them as failures would manufacture a decay that is really just missing time.

Worked example

A cohort of ten customers acquired in January, at a June cutoff:

TENURE	OBSERVABLE	SURVIVING	RATE	
0	10	10	100%	
1	10	9	90%	
2	10	9	90%	
3	10	7	70%	
4	10	7	70%	

TENURE	OBSERVABLE	SURVIVING	RATE
5		10	7
			70%

Concentrated loss at months 1 and 3, then a flat curve — a business losing customers during onboarding rather than at renewal, which points at implementation rather than at product value or price. A single blended “30% churn” figure would have carried none of that.

The choices that change the number

- **The reactivation window** decides whether a gap is churn at all — the single largest lever on every figure on this page.
- **trailing_gap_policy** decides how customers silent at the cutoff are treated. Under the conservative default they count as churned, which makes the most recent months look worse; the alternative makes them look better and can conceal a genuine deterioration in the final periods.
- **Churn timing convention.** First-zero-month (this standard) versus last-active-month shifts every rate by one period.
- **Logo counting after entity resolution.** Whether a subsidiary’s departure is a churned logo depends on where the parent boundary was drawn — see [the measured unit](#).

How it is misread

Annualizing monthly churn linearly. Multiplying a monthly rate by twelve overstates annual churn; compounding is the correct arithmetic, and even that assumes a constant hazard rate the survival curve usually contradicts.

Blending logo and dollar churn. They answer different questions and should never be averaged or quoted interchangeably.

Survival curves computed on all customers rather than observable ones. Produces a manufactured cliff at the right edge of the curve.

Reading the final month or two as a trend. Right-censoring bites hardest there; the trailing-gap policy is doing visible work in those periods, and every affected customer should be read from the register by name.

What it cannot tell you

Customer-level revenue data records that a customer stopped paying. It cannot record *why* — voluntary versus involuntary (payment failure), competitive loss versus budget cut versus acquisition of the customer itself. That split requires churn-reason codes or CRM data joined deliberately, and no amount of care with the revenue file will substitute for it.

Concentration

Concentration measures how much of the business depends on how few customers, on resolved parents, by top-N share and HHI.

ALSO Customer concentration · Top-10 share · Herfindahl–Hirschman Index · HHI

THE STANDARD

Concentration is measured on **MRR at month end**, over resolved parent entities, comped excluded. Reported as top-10 share, top-20 share, a Herfindahl–Hirschman Index over revenue shares, and the trend of all three.

FOR IMPLEMENTERS

SWITCH

DEFAULT

`thresholds.top10_share_warn`

`0.40`

`thresholds.top20_share_warn`

`0.60`

`thresholds.concentration_hhi_warn`

`0.15`

What it measures

Concentration is a measure of *fragility*: how much of the business depends on how few relationships. It is the reason two companies with identical ARR, growth and retention can carry entirely different risk.

Top-N share is intuitive and answers “how exposed are we to the largest few?” HHI is less intuitive and answers something top-N cannot: **how is revenue distributed across the whole book?** Two businesses can share an identical top-10 share while one has a smooth tail and the other a cliff, and HHI separates them because every customer contributes.

The resolved-entity requirement is not a technicality here. Concentration computed on billing identifiers systematically *understates* risk, because the subsidiaries and product-level accounts of one large relationship are counted as separate customers. It is the metric most flattered by skipping entity resolution.

How it is computed

At the measurement month, take every active customer's MRR (comped excluded), sorted descending.

- **Top-N share** = Σ MRR of the largest $N \div$ total MRR. Reported for $N = 10$ and $N = 20$.
- **HHI** = Σ (each customer's share of total MRR)². Ranges from near 0 (a very large number of equal customers) to 1.0 (a single customer). A book of n equal customers has $HHI = 1/n$, which gives the index an intuitive reading: **1/HHI is the "effective number of customers"** — the count of equal-sized customers that would produce the same concentration.
- **Trend** — both shares and HHI computed for every month, so the direction is visible. Direction usually matters more than level: a business whose top-10 share is falling as it scales is de-risking, whatever the absolute number.

Each customer's ranked line also carries its tenure, which distinguishes a large customer of six years from a large customer of six months.

Worked example

Two books, each with total MRR of 100,000 and a top-10 share of 60%:

	BOOK A	BOOK B
Largest customer	12,000	45,000
Customers 2–10	~5,300 each	~1,700 each
Remaining tail	40 customers	40 customers
HHI	0.043	0.222
Effective customers	~23	~4.5

Identical top-10 share; entirely different risk. Book B loses 45% of its revenue if one relationship ends. Top-N alone would have called them equivalent — which is why HHI is reported alongside, and why the threshold of 0.15 flags Book B and not Book A.

The choices that change the number

- **The measurement month.** A spot measure. A month containing an unusual billing event distorts it; the trend view exists to prevent that from being read as a change in structure.

- **MRR versus total revenue.** Measured on MRR here. A business with concentrated *services* revenue may look diversified on this measure while carrying real customer concentration in its total P&L — worth checking separately when services are material.
- **Entity resolution boundaries.** As above: the dominant lever, and the one most often unexamined.
- **Comped exclusion.** Comped customers would otherwise dilute shares while contributing no revenue.

How it is misread

Top-10 share quoted without HHI. Blind to tail structure, as the worked example shows.

Concentration on unresolved billing accounts. Understates risk, always.

Level read without trend. A 45% top-10 share that was 70% two years ago describes a business actively diversifying. The same 45% rising from 30% describes the opposite. The number is identical.

Ignoring tenure. Concentration in long-tenured, contractually committed relationships is a materially different risk from concentration in recent wins — and neither the share nor the index distinguishes them, which is why the ranked table carries tenure.

What it cannot tell you

Concentration measures dependence, not fragility of the dependence. A single customer at 40% of revenue on a five-year contract with switching costs is a different exposure from 40% on a monthly rolling agreement, and the index cannot see the difference. Nor does it capture *correlated* concentration — twenty independent customers in one sector, on one platform, or serving one end-market will behave as a bloc that the customer-level index reads as diversified.

Growth accounting

Growth accounting shows how a business grew, not just how fast: the quick ratio and the mix of new, expansion and reactivation.

ALSO Quick ratio · SaaS quick ratio · Growth mix

THE STANDARD

Growth accounting decomposes each period's MRR change into its five movements and reports the **quick ratio** — gross additions divided by gross losses — together with the mix of additions by source.

FOR IMPLEMENTERS

None of its own; inherits the movement taxonomy and its upstream choices.

What it measures

A growth rate is a single number that conceals its own composition. Growth accounting restores the composition: it asks not *how fast* a business grew but *how it grew*, and whether the way it grew is repeatable.

Quick ratio = (new + expansion + reactivation) ÷ [contraction + churn].

The interpretation follows directly from the arithmetic rather than from any benchmark:

- **Exactly 1.0** — the business adds precisely what it loses. Every unit of new revenue replaces a lost one; growth is zero, whatever the sales effort.
- **Above 1.0** — additions exceed losses; the business grows net.
- **Below 1.0** — the business is shrinking even while selling.

The ratio's value is that it is scale-free and pace-free. A company adding 10 and losing 2 (ratio 5.0) and one adding 100 and losing 20 (also 5.0) have the same growth *efficiency* at different scales. Two companies with identical growth rates and quick ratios of 5.0 and 1.2 have entirely different futures: the second is running a treadmill, and its growth stops the moment sales slows.

Mix decomposes gross additions into new, expansion and reactivation shares. A business growing on expansion has a different cost structure and a different risk profile from one growing on new logos — expansion is generally cheaper to acquire and concentrates the revenue base, while new-logo growth is more expensive and diversifies it. Neither is superior; they are different machines, and a change in the mix over time is often the earliest visible sign that a go-to-market motion is changing.

How it is computed

For each period (quarterly by default, following the fiscal calendar):

- 01 Sum each movement type across the period, signed as in the bridge.
- 02 Gross additions = new + expansion + reactivation.
- 03 Gross losses = |contraction + churn|.
- 04 Quick ratio = additions ÷ losses. Where losses are zero the ratio is undefined and is reported as such rather than as infinity or a large number.
- 05 Mix = each addition type ÷ gross additions.

The same movements that produce the bridge produce these figures, so the two views cannot disagree.

Alongside, the **constant-currency** comparison restates the period at a fixed base-period exchange rate, isolating how much of the reported change was operating and how much was translation. See [FX](#).

Worked example

Two quarters for the same business:

	Q1	Q2
New	100	60
Expansion	40	90
Reactivation	10	10
Contraction	(20)	(25)
Churn	(30)	(35)
Net change	+100	+100

	Q1	Q2
Quick ratio	3.0	2.67
Mix: new	67%	38%
Mix: expansion	27%	56%

Identical net growth. Entirely different quarters. Q2's growth came predominantly from the installed base while new-logo acquisition fell 40%, and losses grew. A headline "we grew 100 again" is true and uninformative; whether Q2 is good news depends on whether the shift to expansion was deliberate.

The choices that change the number

- **Period length.** Monthly quick ratios are volatile; annual ones smooth away the turning points. Quarterly is the default compromise.
- **Reactivation in the numerator.** Included here as a gross addition. Excluding it lowers the ratio and treats win-backs as recoveries of a prior loss rather than as new revenue — defensible, and a different number.
- **The reactivation window,** again: a longer window removes both a churn and a reactivation from the same period, raising the ratio by shrinking the denominator.
- **FX treatment.** Currency-driven movements inflate both additions and losses in a multi-currency book. The constant-currency view removes them.

How it is misread

Quick ratio quoted without the mix. A high ratio driven entirely by expansion in a business with no new logos is a ceiling approaching, not health.

Comparing quick ratios across period lengths. A quarterly and an annual ratio are not comparable; longer periods net more offsetting movements internally.

Treating an undefined ratio as excellent. Zero losses in a period usually means a small base or a short history, not perfection.

Ignoring reactivation's double-count risk. If reactivations are counted as new *and* the original churn stays in the denominator of an earlier period, the same customer has been counted as a loss and as an acquisition. The movement taxonomy prevents this; ad-hoc spreadsheet versions frequently do not.

What it cannot tell you

Growth accounting is silent on cost. A quick ratio of 4.0 achieved with unsustainable acquisition spend and one achieved through organic word of mouth are identical here. It is also silent on quality of revenue:

expansion won by discounting shows as expansion. Pair it with unit economics, which require cost data this standard does not assume.

[HTTPS://WWW.STRATEGYOFFINANCE.COM/STANDARDS/GROWTH-ACCOUNTING/](https://www.strategyoffinance.com/standards/growth-accounting/) · SOF-V1.0

ACV and ARPA

ARPA is average revenue per account. The standard reports the whole distribution, because the average describes almost no customer.

ALSO Average revenue per account · ARPA · Annual contract value · ACV bands

THE STANDARD

ARPA is total MRR divided by active customers in the month. Because the average alone is nearly always misleading, the standard reports the **distribution**: per-customer annualized revenue at every decile, and customer counts and revenue by ACV band.

FOR IMPLEMENTERS

SWITCH

DEFAULT

segments.acv_bands

5K / 25K / 100K / 500K

LOCKED

upper bounds in base currency, annualized

What it measures

ARPA — average revenue per account — is the simplest metric here and the most frequently over-read. Its value is not the level but the *trend* and, above all, the *distribution behind it*.

Software revenue distributions are almost never symmetric. A book with a handful of large enterprises and a long tail of small accounts produces an average that describes no actual customer: it sits far above the median, and moves when one large customer arrives or leaves. Reporting the average alone invites the reader to imagine a typical customer who does not exist.

Two separate ARPA readings answer different questions:

- **Installed-base ARPA** — total MRR ÷ active customers. What the average existing relationship is worth.
- **New-business ARR per logo** — annualized new MRR ÷ new customers in the month. What the business is *currently selling*.

Their divergence is one of the most informative signals in the report. New business consistently landing above installed-base ARPA means the company is moving upmarket, and the average will rise over time. Consistently below means it is moving down — often invisible in the headline ARPA for a year or more, because the installed base changes slowly.

How it is computed

ARPA, per month: $\text{total MRR (comped excluded)} \div \text{count of customers with MRR above zero}$. Undefined and reported as such when no customers are active.

New-business ARR per logo, per month: $\Sigma \text{ new-movement MRR} \times 12 \div \text{count of new movements}$. Reported only for months with new business.

Deciles at cutoff. Take each active customer's $\text{MRR} \times 12$, sort ascending, and report the value at each tenth. The p50 is the median customer; the p90/p50 ratio is a compact measure of skew.

ACV bands. Each active customer is placed in a band by annualized revenue, using the configured upper bounds, and both the customer count and the total MRR in each band are reported. The two together are the point: a band holding 70% of customers and 8% of revenue tells a different story from the reverse, and both appear in every real book.

Worked example

Twenty customers, 100,000 total MRR — ARPA 5,000:

BAND (ANNUALIZED)	CUSTOMERS	MRR	SHARE OF MRR
< 5K	8	2,000	2%
5K–25K	7	8,000	8%
25K–100K	3	15,000	15%
100K–500K	2	75,000	75%

ARPA is 5,000; the **median** customer is around 1,000/month. Fifteen of twenty customers — 75% of the logos — produce 10% of the revenue. The average describes none of them.

This is also the concentration picture from a different angle: two customers carry three quarters of the business. Read together with [concentration](#), the two tables agree, which is a useful consistency check.

The choices that change the number

- **segments.acv_bands** . Band boundaries are conventions. Moving them redraws the picture without changing a single customer, so bands should be set once to match how the business actually segments and then left alone.
- **ARPA on MRR versus total revenue.** Computed on MRR here. Including services raises ARPA and makes it lumpier, since services are episodic.
- **Active-customer definition.** MRR above zero. A customer paused under the reactivation window is carried at held MRR and counts as active — consistent with everything else in the standard, and worth remembering when reconciling against a system that counts differently.
- **Annualization of new business.** New ARR per logo multiplies a single month of new MRR by twelve. In a business selling multi-year contracts with ramps, this understates the eventual value of a new customer, which committed ARR addresses.

How it is misread

ARPA read as the typical customer. The median and the deciles exist for this reason; in a skewed book they can differ from the mean by an order of magnitude.

ARPA growth read as pricing power. ARPA rises when small customers churn, with no price change and no expansion. Always decompose: is ARPA up because existing customers pay more, or because cheap ones left? The bridge answers it; ARPA alone cannot.

Band migration mistaken for growth. A customer crossing a band boundary changes the band table without changing total revenue.

New-business ARR per logo from a thin month. Two new customers in a month make the figure meaningless. Read the count alongside.

What it cannot tell you

Neither measure carries any information about cost to serve or cost to acquire. A high-ARPA enterprise book with long sales cycles and heavy implementation may be less profitable than a low-ARPA self-serve one. ARPA is a revenue-side measure only, and pairing it with acquisition cost requires data outside the revenue file.

Foreign exchange

Revenue converts at the month's average rate, and currency-driven movements are tagged so the constant-currency view can remove them.

ALSO Constant currency · FX translation · Multi-currency revenue

THE STANDARD

Flows are converted at **monthly-average rates** for the month in which they occur. A missing rate falls back to the nearest prior month's rate and is logged as an assumption. Movements arising purely from rate changes are tagged `fx_driven`, sub-totalled in the bridge, and removed entirely in the constant-currency view.

FOR IMPLEMENTERS

SWITCH

DEFAULT

`fx.flow_rate`

`monthly_average`

LOCKED

`fx.fallback`

`nearest_prior`

alternative `block`

What it measures

In a multi-currency book, reported growth is two things fused: what the business did, and what the currency did. Separating them is not an accounting nicety. A company reporting 8% growth in a year when its main billing currency strengthened 10% against the base may have shrunk in operating terms, and no amount of attention to the revenue file will reveal it without an explicit FX treatment.

The standard therefore holds *both* readings: the reported bridge, which is what actually happened to the money, and the constant-currency bridge, which is what happened to the business.

How it is computed

Rate basis. Monthly-average rates, applied to the month a flow occurs. This is the appropriate convention for flows — revenue earned across a month is not translated at a single day's spot rate. Balance-sheet closing rates are a different convention for a different purpose and are not used here.

Native amounts are preserved. Every customer-month row retains its original-currency amounts per measure alongside the converted figures. This is what makes constant currency a recomputation rather than an estimate: the base-currency series can be rebuilt at any chosen rate without returning to the source file.

Missing rates. Under `nearest_prior` (default), the most recent earlier month's rate is used and an assumption is logged naming the currency and month. Under `block`, the computation stops until the rate is supplied. The default trades a small, disclosed imprecision for the ability to compute; the alternative exists for engagements where no inferred rate is acceptable.

FX-driven movements. When a customer's native-currency amounts are unchanged but the base-currency figure moves, the resulting expansion or contraction is tagged `fx_driven`. It remains a movement — the reported bridge must tie to reported revenue — but it is sub-totalled separately so a reader can see how much of the period's expansion was translation.

Constant currency. The entire MRR series is recomputed with every native amount converted at the *base period's* rates, and movements are re-derived from that series. The base period is twelve months before the cutoff by default. With fewer than thirteen months of data the base period predates the data, and every year-over-year constant-currency figure is reported as **not computable** rather than as zero or as a ratio against an empty base.

Worked example

A customer billing 1,000 EUR per month, unchanged, while EUR/USD moves:

MONTH	EUR	RATE	USD MRR	MOVEMENT
Jan	1,000	1.10	1,100	—
Feb	1,000	1.15	1,150	expansion +50 <code>fx_driven</code>
Mar	1,000	1.05	1,050	contraction -100 <code>fx_driven</code>

The customer did nothing. The reported bridge shows expansion then contraction — correctly, because reported revenue genuinely moved. The constant-currency view at January rates shows 1,100 in all three months and no movements at all.

Both are true. Quoting only the first attributes currency to the sales team; quoting only the second describes revenue nobody received.

The choices that change the number

— `fx.flow_rate`. Monthly average is locked. Spot-at-month-end would make every month's revenue depend on one day's rate.

- **fx.fallback** . **nearest_prior** versus **block** — compute with a disclosed approximation, or refuse until the data is complete.
- **The constant-currency base period.** Twelve months back by default. Any other choice produces a different “constant” currency, so the base period is stated on the table.
- **Whether FX movements are movements at all.** Here they are, tagged. Excluding them from the bridge entirely would break the tie to reported revenue.

How it is misread

Constant currency quoted without the base period. Meaningless without it — the phrase names a method, not a number.

FX-driven expansion counted as commercial expansion. Inflates NRR and the quick ratio in exactly the periods when a currency moved favourably. The sub-total exists to prevent this.

Single-rate-for-the-year conversion. Common in hand-built models and wrong in both directions: it misstates individual months and, because the growth comparison spans two different rate environments, misstates the trend too.

Assuming a single-currency file has no FX exposure. A company billing entirely in USD to European customers carries the exposure in its customers' willingness to pay, not in its revenue file. That risk is real and invisible here.

What it cannot tell you

The standard converts recorded amounts at recorded rates. It cannot tell you whether prices were repriced in response to currency moves, whether hedges exist, or whether a customer's local-currency price is sustainable after a large move. It measures translation, not economic exposure.

Revenue tie-out

The customer file counts only once its recognized revenue agrees with the P&L, month by month, with billing timing shown separately.

ALSO Tie-out · Revenue reconciliation · Billed vs recognized vs reported

THE STANDARD

Every report reconciles **recognized** revenue to **reported** revenue, month by month, and separately bridges **billed to recognized** so that timing differences are visible rather than netted. A report built from a file that does not tie to the financial statements is not complete under this standard.

FOR IMPLEMENTERS

SWITCH

`thresholds.tieout_gap_warn`

DEFAULT

`0.02`

2% relative

What it measures

Every metric in this standard is computed from a customer-level revenue file. That file is an operational export. It has no inherent authority: it was not audited, it may exclude entities, it may include revenue the accounts treat differently, and it may simply be incomplete.

Tie-out is the step that gives the file standing. It asks: **does the sum of the customer-level data agree with the revenue the company reports?** If it does not, every downstream metric — ARR, retention, cohorts, concentration — is computed on a population that does not match the business being discussed, and the size of the mismatch bounds how much any of them can be trusted.

This is the check most often skipped in self-prepared metrics reports, and the first one an experienced diligence reader turns to. Skipping it does not remove the discrepancy; it removes the disclosure.

How it is computed

Three quantities per month, from the same customer-month data:

- **Recognized** — the rev-rec view: recurring revenue with prepaid contracts spread over their term and prorated months as billed, plus usage, services, one-time, transactional, credits and other non-recurring. This is the figure that should agree with the P&L.
- **Billed** — raw billed amounts at the period they were invoiced, lump sums intact.
- **Reported** — monthly revenue as stated by management, supplied separately.

From these:

- **Gap** = reported – recognized, in currency and as a percentage of reported. Months where the absolute relative gap exceeds `thresholds.tieout_gap_warn` are flagged.
- **Billing timing** = billed – recognized. Not an error: it is prepayment and proration, and it should be large in a business that bills annually in advance. A billing-timing line near zero in an annual-billing business is itself a signal that lump sums were not spread.
- **Pause-held MRR** — MRR held through qualifying pauses, which by design appears in the run-rate series and *not* in recognized revenue. Isolating it explains a gap between run-rate and rev-rec views that would otherwise look like an error.

Where no reported financials are supplied, the gap column is empty, the table says so plainly, and the limitation is recorded in the register. The tie-out is never silently omitted.

Worked example

MONTH	RECOGNIZED	REPORTED	GAP	GAP %	BILLED	TIMING
Jan	100,000	100,000	0	0.0%	340,000	+240,000
Feb	102,000	102,000	0	0.0%	12,000	–90,000
Mar	104,000	118,000	14,000	11.9%	105,000	+1,000

January and February are healthy: recognized ties exactly, and the large billing swing is annual invoicing being spread correctly.

March is the finding. A 14,000 gap at 11.9% means the P&L contains revenue the customer file does not — a business line excluded from the export, a non-customer revenue stream, an acquired entity not yet in the operational system, or a manual accounting adjustment. Until it is explained, every March metric rests on 88% of the actual revenue base, and the report must say so.

The choices that change the number

- **Which reported figure is used.** Total revenue, or recurring revenue only. Comparing recognized (all types) against a recurring-only reported figure manufactures a gap the size of the non-recurring business.
- **Contract data availability.** Without contracts, prepaid lump sums cannot be spread and recognized approaches billed — which widens the gap against a P&L that does spread them. The absence of contracts is thus visible *in the tie-out*, not only in the register.
- **Cutoff alignment.** A data cutoff mid-period against a reported figure for a full period produces a gap that is purely a boundary artifact.

How it is misread

A gap treated as an error to be plugged. The gap is a finding. Adjusting the data to match the P&L destroys the only independent check in the report.

Billing timing read as a discrepancy. It is the expected consequence of annual billing and should be large; the question is whether its *pattern* matches the stated billing terms.

A perfect tie treated as complete validation. Tie-out proves the file's totals agree with reported totals. It says nothing about whether revenue is attributed to the right customers or the right months *within* those totals — a file with two customers' revenue swapped ties perfectly and produces entirely wrong retention.

No tie-out at all, presented as a clean report. The absence is the finding.

What it cannot tell you

Tie-out compares the operational file to management's reported figures. It cannot validate the reported figures themselves — that is what an audit is for — and it cannot detect an error that exists identically in both. It establishes *consistency*, which is a necessary condition for trusting the metrics and not a sufficient one.

Definitional sensitivity

The same data yields a family of NRRs. The spread between them is a finding about the business, not an error.

ALSO NRR sensitivity · Definition spread · Retention variants side by side

THE STANDARD

A report under this standard computes its headline retention under **all** defined variants and toggles — same data, same period, only the definition changes — and reports the spread between them in percentage points. A wide spread is not an error to be resolved. It is a property of the business, and it is a finding.

FOR IMPLEMENTERS

SWITCH

`thresholds.nrr_spread_warn_points`

DEFAULT

5

What it measures

Every other entry in this reference describes one measure and names the choices that move it. This one measures the moving.

The premise is uncomfortable and worth stating plainly: for most businesses, there is no single true NRR. There is a family of defensible NRRs, and which one gets quoted is a choice. Ordinary metrics reporting hides that choice by presenting one number. Sensitivity analysis inverts it — it presents the whole family and lets the spread carry information.

A narrow spread means the headline is robust. Whatever definitional argument a counterparty brings, the answer barely moves. That is a strong position, and the company should know it before the meeting.

A wide spread means the headline is definition-fragile. The number depends more on the convention than on the business. This is not an accusation — usage- heavy businesses and those with irregular billing are structurally fragile here — but it means any single quoted figure is a claim about methodology as much as about performance, and diligence will find the least flattering variant.

Either way, the company learns what a buyer will compute before the buyer computes it. That is the entire purpose.

How it is computed

At the cutoff, the same customer-month data is measured under every combination the standard defines. For NRR: TTM compounded (the default), TTM usage-exclusive, TTM reactivation-exclusive, year-over-year snapshot usage-inclusive and usage-exclusive, and quarterly annualized. For GRR: the corresponding set.

The default variant is marked. The **spread** is the maximum NRR minus the minimum, in percentage points, and a spread exceeding `thresholds.nrr_spread_warn_points` raises a finding.

Because every row runs against the identical data over the identical period, the only difference between any two rows is the definition. Nothing else can explain the gap.

Worked example

DEFINITION	NRR
TTM compounded · usage-incl · reactivation-incl (<i>default</i>)	112.4%
TTM compounded · usage-exclusive	98.1%
TTM compounded · reactivation-exclusive	109.7%
YoY snapshot · usage-inclusive	104.2%
YoY snapshot · usage-exclusive	91.6%
Quarterly annualized	118.9%
Spread	27.3 points

One business, one period, one dataset. The company can honestly say “our NRR is 112%.” A buyer can equally honestly say “your NRR is 92%.” Both are computing correctly, and the argument that follows is not about the data at all.

The instructive comparison is the usage pair: 14 points between usage-inclusive and usage-exclusive on the same variant. That single fact — that consumption revenue carries a seventh of this company’s apparent retention — is more informative than any individual row, and it is invisible in a report that gives one number.

The choices that change the number

The spread is *made of* choices; the meta-choices are:

- **Which variants are in scope.** This standard fixes the set so the spread is comparable across reports. Curating the set — dropping the variant that reads worst — would defeat the exercise.
- **The warning threshold.** Five points by default. A convention for flagging, not a judgment about the business.
- **Whether to publish it at all.** The standard's answer is that it is not optional. A table that exists to show fragility cannot be suppressed when the fragility is high without destroying the guarantee it provides.

How it is misread

A wide spread read as a data-quality problem. It usually is not. It reflects genuine composition — usage mix, reactivation frequency, seasonality — and cleaning the data will not narrow it.

The most favourable variant selected for the headline. The table makes this visible, which is its point. A company that quotes its best variant *while publishing all of them* is making a disclosed choice; one that quotes it while publishing nothing else is making an undisclosed one.

The spread compared across companies as a quality score. Spread is driven by business model. A pure subscription business will have a narrow spread almost regardless of how well it is run.

What it cannot tell you

Sensitivity analysis cannot tell you which definition is *right*. That depends on the question being asked, and reasonable parties will differ. What it guarantees is that nobody is surprised — that the range is known to both sides before the negotiation, and that the discussion is about which convention fits rather than about whose spreadsheet is correct.

The register

Every assumption applied and every anomaly found is logged, so a reader can argue with a premise instead of discovering it later.

ALSO Assumptions register · Exceptions register · Assumptions and exceptions log

THE STANDARD

Every gap in the data produces a logged, visible, reviewable entry. Assumptions record where a default was applied; exceptions record where the data was anomalous. The register accompanies every report. A report without its register is not complete under this standard.

FOR IMPLEMENTERS

Every threshold in the definition config feeds a register finding.

What it measures

The register measures the distance between the data the company has and the data the metrics assume.

Every real revenue file has gaps: rows with no revenue type, currencies with no rate, customers with billing holes, duplicates, negative amounts with no explanation, contracts absent for prepaid billings. Something must be done with each. The alternatives are to guess silently, to stop entirely, or to **apply a stated default and record it**. The first destroys trust the moment it is discovered; the second makes the analysis impossible, since no file is ever clean.

The register is the third path, and it carries a claim that the rest of the report depends on: *these are the assumptions, all of them, and here is what each one did to the numbers*. An assumption a reader can see is a premise they can argue with. An assumption they cannot see is a defect waiting to be found by the party least sympathetic to the company.

How it is computed

Two kinds of entry, both stable-keyed so they can be reviewed, tracked, and compared across restatements.

Assumptions — a default was applied where the data was silent:

- the declared amount basis and period grain of each file
- revenue-type fallbacks, and per-product type rules
- pause holds, naming the customer, the gap, and the held amount
- FX fallbacks, naming currency and month
- trailing-gap policy applications
- coercions performed during parsing
- label variants differing only by case or spacing, treated as distinct
- multi-sheet workbooks, naming the sheet read and those unread

Each carries what was applied, the alternatives, and whether it is an automatic default or a decision confirmed by a person.

Exceptions — the data itself was anomalous: duplicate rows, billing gaps, negative amounts, proration anomalies, orphan entities, missing rates, unknown revenue types, suspected related parties, indeterminate churn at the cutoff, ambiguous grain, parse failures. Each carries a severity — `info`, `warn`, or `blocking` — and its scope.

Blocking is real. Some conditions stop the computation rather than degrading it: hybrid revenue with no type column, an amount basis that cannot be determined. The standard refuses to produce a number over an input that requires a guess about what the numbers *mean*.

Review. Each entry can be marked by a reviewer as **accepted** (understood, stands as logged), **disputed** (the logged item looks wrong), or **resolved** (fixed at the source), with a note. Reviews are append-only and travel with the report, so the register is a working paper — the record of a conversation between the people who know the data and the people who must rely on it — rather than a static list of complaints.

Worked example

A register entry from an ordinary file:

```
Assumption · pause_hold:ent_acme:2024-04 · category pause_hold  
· status auto Description: Acme Corp: no revenue for 2 month(s)  
from 2024-04, resuming within the 3-month reactivation window —  
treated as a pause; MRR held at 5,000 per the SoF reactivation-window  
rule. Default applied: MRR held through gap; no churn/reactivation  
recorded. Alternatives: Treat as churn + reactivation (shorten window).
```

A reader who disagrees knows exactly what to change, which customer is affected, and what the alternative produces. Compare that to the same decision made silently inside a spreadsheet formula, where the only way to find it is to already suspect it.

The choices that change the number

The register does not change any number — it *describes* the changes other choices made. That is precisely its function: it is the audit trail for every configurable decision documented in the other sixteen entries.

The one genuine choice here is **completeness**. A register that omits inconvenient entries is worse than no register, because it carries an implicit claim of exhaustiveness. Under this standard the register cannot be edited or filtered — only reviewed.

How it is misread

Read as a defect list. A long register does not mean bad data; it usually means thorough disclosure. A suspiciously short one on a real-world file means the checks were not run.

Counts read without severity. Forty **info** entries and one **blocking** entry are not comparable, and the summary counts exist to be read by category.

Treated as optional in a summary report. The register is precisely what makes the summary credible; removing it for brevity removes the reason to believe the rest.

Reviewed once and forgotten. Restating with new data regenerates entries. Reviews attach to stable keys so prior decisions persist, but new entries need new attention.

What it cannot tell you

The register records what the *computation* encountered. It cannot record what is absent from the file entirely — a business line never exported, a subsidiary excluded from the extract, revenue booked outside the operational system. That class of gap is only visible by comparison against the financial statements, which is why tie-out is a separate and mandatory check.

A report under this standard vouches for the computation over disclosed inputs. It does not vouch that the inputs are the whole truth — no analysis of a file can do that, and any claim otherwise should be treated as the strongest reason to distrust it.