

METERING, BILLING, SETTLEMENT & REVENUE PLATFORM

Engineering Specification v1.0 — the capacity (GPU-hour) and token (inference) businesses

Prepared for Reliance Intelligence platform engineering. Companion build artifacts — JSON Schemas, OpenAPI 3.1, PostgreSQL and ClickHouse DDL, 17 golden test vectors, executable reconciliation SQL, and a conformance checklist — ship alongside this document and are the normative contract where the two disagree.

Status: draft for engineering review. Section 16 lists the commercial and tax decisions required before build starts. Figures are illustrative; India tax and accounting treatments must be confirmed by Reliance's tax and statutory audit teams before go-live.

1. Purpose, Scope, and Non-Goals

This platform converts physical events at Jamnagar — a GPU allocated, an inference request served — into money that has been correctly measured, correctly priced, correctly invoiced, correctly collected, and correctly recognised. It is the system of record for revenue. Nothing else in the estate is permitted to compute a customer-facing amount.

The scope covers two commercially different products running on one physical fleet. The capacity business sells GPU-hours, mostly under multi-year take-or-pay reservations, with on-demand and spot monetising what the committed book leaves. The token business sells inference metered per million tokens, with distinct rates for input, cached input, and output, across interactive, flex, batch, and provisioned tiers. They share a fleet, a cost base, and an assurance discipline, but almost nothing about their metering: one meters an interval, the other meters an event, and conflating them is the most common architectural mistake in this domain.

1.1 In scope

- **Metering** — event capture from the control plane, node agents, API gateway, and serving engines; idempotent ingestion; quarantine; late-event handling.
- **Rating** — deterministic, reproducible conversion of usage to money against versioned price books, including tiering, commitment drawdown, and index-linked merchant rates.
- **Invoicing** — invoice assembly, gapless GST-compliant numbering, e-invoice (IRN) registration, credit and debit notes, delivery.
- **Settlement** — cash application including customer TDS withholding, export receipt matching (FIRC/e-BRC), dunning, and disputes.
- **Revenue** — Ind AS 115 performance obligations, deferred revenue, unbilled receivable, variable consideration, and the RPO/backlog snapshot.
- **Assurance** — the three-way match, leakage taxonomy, control library, and KPI set.

1.2 Explicitly not in scope

The platform does not schedule workloads, does not decide who gets capacity, and does not set prices — it consumes the control plane's allocation decisions and the deal desk's published price books. It is not the general ledger; it produces journal-ready output and posts to the GL. It is not the CRM or CPQ; contracts arrive as structured data from those systems. And it does not store prompt or response content: the billing plane needs counts and identifiers, never the text, and that boundary is a data-protection control as much as an architectural one.

2. Design Principles

Six rules constrain every design decision that follows. They are stated here because most of the specific requirements later in the document are consequences of them, and because a team that internalises the rules will make the right call on the hundred decisions this document does not anticipate.

1. **Events are immutable** — A usage event, once accepted, is never updated and never deleted. Corrections are new events carrying `correction_of` and a reason, with signed quantities. This is what makes it possible to answer "what did we know, and when?" three years later, which is the only question that matters in a dispute or an assessment.
2. **Rating is a pure function** — Given the same events and the same pinned price-book version, the engine produces byte-identical output, forever. Every rated line stores an `input_hash` over its inputs. An invoice that cannot be reproduced cannot be defended.
3. **No floating point** — Money is integer minor units (paise). Rates and quantities are exact decimals, transported as strings. IEEE-754 floating point is banned from the money path and the ban is enforced by a CI lint, not by convention. Rounding is half-even, applied once per rated line, and document round-off is applied once at the invoice total.
4. **Bill from intent, reconcile against reality** — For capacity, the control-plane allocation ledger bills and node telemetry checks. For tokens, the API gateway bills and the serving engine checks. Two independent legs, or the reconciliation is theatre — a single source reconciled against itself always balances and always lies.
5. **Never drop, always quarantine** — An event that cannot be rated is quarantined with an owner and an SLA. It is never dropped. A dropped event is revenue that no control will ever find, because nothing remains to find it with.
6. **Contract terms are data** — Every commercial term that affects an amount — a tier mode, a cache-hit rate, a discount, a forfeiture — is data in a versioned price book or contract, not a branch in code. Terms change weekly; deployments should not have to.

3. Architecture

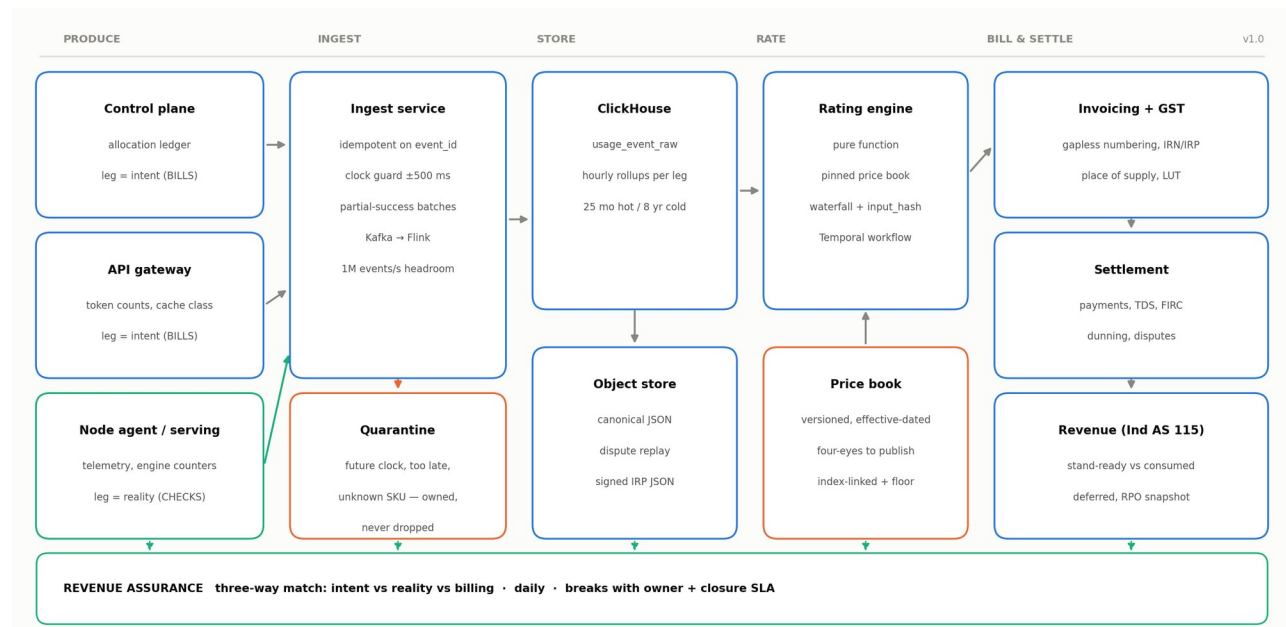


Figure 1 — Component architecture. Blue components are on the billing path; orange are governance and exception paths; green is the independent reality leg and the assurance plane that spans everything.

The shape is a pipeline with two governance interruptions. Usage flows left to right: producers emit, ingest deduplicates and guards, the event store aggregates, the rating engine prices, and the billing layer invoices, collects, and recognises. The two interruptions are the quarantine (events that cannot be rated stop and wait for a human) and the price book (rates cannot enter the engine without a second approver). Underneath, the assurance plane reads all three legs independently and opens breaks — it is deliberately not in the pipeline, because a control that depends on the system it audits is not a control.

Two properties of this layout matter more than the component choices. First, rating reads aggregates but every rated line retains the identifiers needed to walk back to raw events, so the evidence path is intact end to end. Second, the billing ledger and the event store are separate databases with separate failure domains: an analytics query storm cannot threaten invoice integrity, and an event-store outage degrades new rating without endangering documents already issued.

3.1 Reference stack

The stack below is a recommendation with rationale, not a mandate. Each row names an alternative that a platform team could reasonably prefer; what is not negotiable is the property in the third column, because the design depends on it.

Layer	Recommendation	Required property	Alternative
Event transport	Apache Kafka (MSK or self-managed)	Durable, ordered per key, replayable ≥ 7 days	Redpanda; Apache Pulsar
Stream processing	Apache Flink	Exactly-once sinks, event-time windows, late-data handling	Kafka Streams; Spark Structured Streaming
Event store	ClickHouse	Column store, 100B+ rows, tiered hot/cold storage	Apache Druid; StarRocks
Billing ledger	PostgreSQL 16 (Patroni HA)	ACID, FK integrity, row locks for numbering	CockroachDB (multi-

Layer	Recommendation	Required property	Alternative
			region); YugabyteDB
Orchestration	Temporal	Durable long-running sagas, deterministic replay	Cadence; (Airflow is a poor fit — it is a scheduler, not a saga engine)
Object storage	S3-compatible on-prem (MinIO/Ceph)	Immutable object lock, 8-year retention, in-country	Any S3-compatible store inside India
Cache / locks	Redis (Sentinel or Cluster)	Idempotency-key cache, rate limits	Valkey; DragonflyDB
Services	Go or Java 21, gRPC internal / REST external	Exact decimal libraries; no implicit float coercion	Rust; Kotlin
Schema registry	Confluent Schema Registry (Protobuf)	Backward-compat enforcement at publish time	Apicurio; Buf Schema Registry
Secrets & signing	HashiCorp Vault + HSM	DSC for e-invoice signing held in hardware	Any FIPS 140-2 L3 HSM
Observability	OpenTelemetry → Prometheus / Grafana / Loki	Per-stage event counts as first-class metrics	Datadog; Elastic

Table 1 — Reference stack. The one strong opinion is Temporal for the billing run: a monthly close is a multi-hour saga with retries, human approvals, and compensations, and hand-rolling that on a scheduler is where billing platforms go to die.

A note on build versus buy

Commercial usage-billing platforms (Metronome, Orb, Lago, Zuora) solve rating, invoicing, and dunning credibly, and a team under time pressure should evaluate them for exactly those layers. What none of them solve is the part specific to this business: capacity metering against a GPU control plane, take-or-pay drawdown at cluster granularity, Indian e-invoicing at Reliance's document volumes, and the three-way match against node telemetry. The pragmatic split is to build metering, aggregation, and assurance — the parts that are Reliance-specific and where the leakage is — and to consider buying rating and invoicing if the schema contracts in this bundle can be preserved at the boundary. If they cannot be preserved, build.

4. Domain Model

Ten entities carry the business. Their relationships are constrained in the DDL rather than in application code wherever a constraint is expressible, because the cost of a bad row in a revenue ledger is measured in audit findings rather than in bug tickets.

Entity	Holds	Key invariant
Tenant	Selling legal entity: GSTIN, PAN, state, SEZ status, LUT	One tenant per GSTIN; LUT expiry blocks zero-rated invoicing
Account	Billed party: tax identity, currency, terms, TDS profile	Expected TDS rate lets AR tell withholding from a dispute
Contract / ContractLine	Commitment quantity, rate, ramp, acceptance, SLA tier, rollover policy	Billing cannot start before accepted_at when acceptance is required
PriceBook	Versioned, effective-dated rate entries with SAC codes	Immutable once published; publisher ≠ author
UsageEvent	One billable action, with provenance and leg	Unique on event_id, forever; never mutated
RatedLine	Quantity × resolved rate, waterfall, input hash, revrec class	net = gross + discount; superseded by re-rate, never edited
Invoice / InvoiceLine	Assembled document, tax, IRN block	Gapless number per (GSTIN, FY, series); immutable once issued
Payment / Application	Receipt and its allocation, incl. TDS withholding	Applications sum to the receipt; unapplied is visible
PerformanceObligation / RevenueSchedule	Ind AS 115 classification and period amounts	Recognised ties to invoiced ± deferred ± unbilled
ReconciliationBreak	A three-way-match discrepancy	Has an owner, a materiality, and a closure date

4.1 The two meters

Capacity is an interval meter. The control plane emits one event per allocation per window (default 300 seconds) carrying `gpu_seconds`, and the node agent emits a matching event on identical window boundaries so the two legs join on (`allocation_id`, `window_start`). Getting those boundaries to agree is a design-time obligation on both producers; if they drift, the reconciliation can never balance and the control becomes noise that teams learn to ignore.

Tokens are a point meter. The API gateway emits one event per completed request, and it — not the serving engine — is the billing leg, because only the gateway sees the request as the customer does: authenticated identity, billing account, cache classification, service tier. The serving engine emits the same counters on the reality leg. The subtlety that causes the most revenue damage is that `cached_input_tokens` must be *disjoint* from `input_tokens`, not nested inside them. A producer that reports the total prompt in both fields bills the cached portion twice at two different rates, and because cache hit rates are high and the cached rate is low, the error is small enough per request to hide in the noise and large enough in aggregate to matter.

5. Metering

5.1 Producer obligations

Producers are the only place revenue can be lost without trace, so their obligations are contractual rather than best-effort. Every producer must generate a deterministic event_id (UUIDv7) so that a retry of the same logical event carries the same identity; must buffer to local durable storage with at least six hours of depth when ingest is unreachable and drain in order on recovery; must run a PTP- or NTP-disciplined clock, with drift beyond 500 ms raising an alarm; and must stamp source.version on every event, so that when a meter defect is found the blast radius can be computed exactly instead of estimated conservatively.

Streaming token requests emit a single event at stream close, including when the client disconnects mid-generation. Tokens produced before a disconnect consumed real GPU time and are billable; tokens lost to a server error are not, and the zero-rated event is retained so that the error rate is measurable from the billing data itself rather than from a separate observability pipeline that may disagree.

5.2 Ingestion

The ingest service accepts batches of up to 5,000 events and returns per-event outcomes. Partial success is the contract, deliberately: a producer forced to drop an entire batch because one element was malformed will drop revenue, and it will do so quietly. Each event is accepted, deduplicated, quarantined, or rejected on its own merits.

```
POST /v1/usage/events:batch      Idempotency-Key: <uuid>

202 Accepted
{ "accepted": 4987, "deduplicated": 11, "quarantined": 2, "rejected": 0,
  "results": [ { "event_id": "...", "status": "quarantined",
                 "reason": "occurred_at 4.2h beyond late-event window" } ] }
```

Idempotency is enforced twice — a seven-day dedupe window in the stream layer and a permanent unique constraint in the ledger — because either mechanism alone will eventually be bypassed by a replay, a backfill, or a migration. The clock guard rejects nothing: events more than 500 ms in the future, or older than the late-event window (default 72 hours), are quarantined for operator disposition. Quarantine is a work queue with an owner and an ageing SLA, not a dead-letter topic that nobody reads; aged quarantine beyond 72 hours pages, because by then it is unrecoverable revenue rather than a backlog.

5.3 Late events and re-rating

Late arrival is normal, not exceptional — a node agent buffering through a network partition will flush after the period closed. The system handles it by design: the late event lands, the affected period is re-rated, the original rated line is superseded (retaining rating.rerate_of) and reversed, and a replacement line is issued. If the invoice for that period has already been issued, the difference becomes a debit or credit note. The invoice is never edited. Because this path runs routinely rather than as an emergency, it is exercised and trusted; the alternative — a manual adjustment process invoked twice a year by whoever is available — is how billing systems acquire unexplainable balances.

6. Storage and Aggregation

Raw events land in ClickHouse in a ReplicatedReplacingMergeTree keyed so that a redelivered event_id collapses on merge rather than double-counting. Partitioning is monthly on occurred_at; the retention policy moves data to cold object storage at 25 months and deletes at 8 years, which satisfies both the Companies Act books-and-

records requirement and GST record retention (72 months from the due date of the annual return). The canonical JSON of every event is retained alongside the parsed columns, because dispute resolution replays the original payload, not our interpretation of it.

Above the raw table sit `AggregatingMergeTree` rollups, one set per reconciliation leg. Rating reads the rollups; the assurance controls read both legs and compare. Keeping the legs in separate rollup rows rather than separate tables means the comparison is a single scan over one ordering key, which is what makes a daily three-way match over a 63,000-GPU fleet finish in minutes rather than hours.

One materialised view deserves specific mention. The tokenizer-drift detector reports any model serving more than one `tokenizer_version` in a day. A tokenizer change re-prices every request by a few percent in one direction, is invisible in daily revenue trend, and is discoverable only by someone reconciling counts — so the detector runs continuously and, per control C-14, blocks the billing close rather than filing a ticket.

7. Rating

7.1 The engine contract

Rating is a pure function from (aggregated usage, pinned price-book version, contract terms snapshot) to rated lines. It is orchestrated as a Temporal workflow so that a multi-hour close survives restarts, and it runs in three modes. Commit produces the lines that become invoices. Dry run produces lines in a shadow namespace with a diff against the last committed run, and a material delta blocks the close until explained — this is the control that prevents a rating change shipping straight into revenue. Re-rate supersedes prior lines after late events or a meter correction.

7.2 The discount waterfall

Every rated line records an ordered waterfall from list price to net, and the order is fixed: list, then tier, then contract rate, then commitment discount, then promotional credit, with index resolution and floor or cap application recorded where applicable. Each step names the authority for itself — a contract line, a promotion identifier, an approval ticket. The waterfall is renderable in the customer portal on request, because a discount that cannot be explained is a dispute waiting to be filed, and an internal team that cannot explain one has no way to detect that it should have expired.

7.3 Arithmetic

The working precision is twelve decimal places. Quantity conversion uses exact rational factors — `gpu_seconds` divided by 3600, tokens divided by one million — computed as a single fused expression with the rate, never as an intermediate rounded quantity. Vector TV-CAP-002 makes the point concretely: on a 1,000-second window at ₹440 per GPU-hour, truncating the quantity to four decimal places first yields 12,223 paise where the correct answer is 12,222. One paise on one line is nothing. Applied across a merchant book of roughly 164 million GPU-hours a year it is real money, and more damagingly it produces a reconciliation break that nobody can explain, which is how organisations learn to ignore their reconciliations.

7.4 Index-linked rates

The price book supports an `index_linked` model for merchant capacity, resolving to $\max(\text{index print} + \text{spread}, \text{floor})$ with an optional cap and an FX conversion where the index publishes in USD. It exists so that physical contracts can reference the same public benchmark the treasury hedges against — the CME Silicon Data H100 and B200 rental index futures listing on NYMEX, or ICE's planned contracts on the Ornn Compute Price Index. Two implementation requirements are load-bearing. The floor is mandatory: it is the margin protection that makes

index-referencing safe for the deal desk to offer at all. And both the index print and the FX rate are frozen onto the rated line, so that a later restatement of the index does not silently rewrite history.

8. Contracts, Commitment, and SLA

Take-or-pay is the commercial spine of the capacity business and the source of its most common billing errors. The invariants are simple to state and easy to get wrong. The commitment bills regardless of consumption. Consumption draws the commitment balance down. Consumption above the commitment bills as overage, on its own rated line, at its own rate — silently absorbing overage into the commitment line is a first-order contract-leakage mode. Consumption below the commitment is forfeited or rolled over according to an explicit per-contract policy, and rollover creates a contract liability with an expiry that the platform must track or revenue is recognised early.

All four positions — consumed, committed, overage, forfeited — are exposed to the customer through the API and the portal. Take-or-pay economics that a customer cannot see are take-or-pay economics a customer disputes. Provisioned-throughput token traffic follows the same principle: it is metered and shown on the invoice zero-rated, with the capacity contract carrying the charge, rather than omitted. A customer who cannot verify what their reservation bought will ask, and the answer should be on the document.

Delivery milestones gate revenue. Where a contract line requires acceptance, no usage before accepted_at is billable, and the RPO report separates the backlog that is convertible from the portion contingent on an acceptance that has not yet happened. On a greenfield campus energising in tranches, that separation is the difference between a backlog number a lender can rely on and one they will discount.

SLA credits flow the other way and are computed from the incident record, not requested by email. A credit must match an incident; the control that checks this (C-06) exists because discretionary credits are the softest channel through which revenue leaks. Critically, expected credits are estimated as variable consideration at recognition rather than deducted only when issued — a provider with chronic incidents should recognise less revenue in the period, not full revenue followed by an embarrassing credit next quarter.

9. Invoicing and Indian GST

Cloud and IT services attract GST at 18%, with SAC 998315 for hosting and infrastructure. The SAC code is carried on every price-book entry rather than derived at invoice time, because it is a commercial classification that the deal desk owns and that the tax engine should not have to guess.

9.1 Numbering

Invoice numbers must be at most sixteen characters, alphanumeric with slash and hyphen only, unique per financial year per GSTIN series, and gapless. Allocation happens at issue, not at assembly, under a row lock on a sequence table — deliberately not a PostgreSQL SEQUENCE, which is non-transactional and leaves gaps on rollback. A failed issue burns the number and records the fact rather than silently reusing it. This is unglamorous and it is the first thing a GST audit examines.

9.2 E-invoicing and IRN

E-invoicing applies to B2B supplies, SEZ supplies, and exports for suppliers above the notified aggregate-turnover threshold (₹5 crore at the time of writing), and covers tax invoices, credit notes, and debit notes. B2C is excluded. The flow is: submit the GST INV-01 JSON to an Invoice Registration Portal, receive a 64-character IRN, a signed QR code, and signed JSON; render the QR on the PDF; archive the signed JSON verbatim.

Two engineering decisions follow. First, IRN registration is asynchronous and off the critical path: an IRP outage must never stop Reliance from issuing invoices. The document reaches issued status once it is persisted and numbered; `irp_status` stays pending and is retried with backoff. Second, the IRP's duplicate-IRN response (error 2150) is treated as success, storing the returned IRN — treating it as an error produces a retry loop that generates a support incident every time a network timeout hides a successful registration.

The thirty-day reporting limit is a commercial deadline, not a technical one. An invoice unreported past thirty days from its issue date is rejected by the IRP, and the customer consequently cannot claim input tax credit. The platform alerts at T+7, pages at T+21, and raises an incident at breach.

9.3 Place of supply, exports, and SEZ

For B2B services the place of supply is the recipient's location, which resolves the supply to intra-state (CGST plus SGST at half the rate each) or inter-state (IGST). Exactly one branch applies; a line carrying both CGST and IGST means the resolver is wrong, and the DDL rejects it at write time rather than letting it reach a return.

Exports of service are zero-rated under section 16 of the IGST Act, invoiced without tax under a Letter of Undertaking (Form GST RFD-11, renewed annually). Zero-rated is not exempt: the invoice still requires an IRN, input credit is preserved, and the receipt must be realised within the FEMA window with an FIRC or e-BRC matched back to the invoice. An expired LUT changes the treatment to export-with-payment, which is a cash-flow event that finance must consent to — so the platform blocks export-without-payment invoicing the day the LUT lapses and warns 45 days ahead, rather than guessing.

If the Jamnagar campus operates as or supplies to an SEZ unit, supply-type codes change (SEZWP and SEZWOP) and certain e-invoicing exemptions apply to SEZ units themselves. This is the single most consequential open item in Section 16: it changes document codes, tax treatment, and exemption logic, and it cannot be inferred from the architecture.

9.4 Amendments

An issued invoice is immutable, enforced by a database trigger rather than by code review. Reductions are credit notes; increases are debit notes; each is itself a GST document requiring its own IRN. Credit reasons are governed,

and only `billing_error` counts against the billing-accuracy KPI — an SLA credit is a commercial outcome, not a billing defect, and conflating the two makes both numbers useless.

10. Settlement

The two product lines settle completely differently and need different machinery. Self-serve token customers settle like consumer SaaS: cards, prepaid credits, processor fees, chargebacks, and a retry ladder for failed payments. Enterprise capacity customers settle like industrial leasing: net-30 to net-60 invoices, wires, occasional letters of credit, and DSO management against a concentrated book where receivables risk is really single-name credit risk and belongs in front of a credit committee rather than an AR clerk.

10.1 TDS withholding

Indian enterprise customers remit net of tax deducted at source — typically 2% under section 194J for technical services. A receipt short by exactly the expected withholding is a lawful payment, not a dispute, and the platform applies it as `withholding_tds` with the residual sitting in a TDS-receivable control account. Getting this wrong opens a false dispute on essentially every enterprise payment, which destroys the credibility of the dispute queue. The loop closes when the Form 16A certificate or the Form 26AS credit is matched; unmatched withholding at year end is a genuine cash loss and belongs on the assurance dashboard (control C-10), not buried in an ageing report.

10.2 Export receipts and disputes

Export receipts are matched to an FIRC or e-BRC within the twelve-month realisation window; breaches are raised as compliance exceptions rather than as slow-paying customers. Disputes carry the raw event identifiers that evidence them, and are settled by replaying those events. A dispute process that negotiates over aggregates concedes revenue it did not have to; one that produces the `request_id` and the timestamps usually ends the conversation.

11. Revenue Recognition (Ind AS 115)

Classification is decided at rating time and stored on the rated line, not derived in a spreadsheet at close. Three treatments cover almost everything.

- **Committed capacity** — a stand-ready obligation. Revenue is recognised ratably as capacity is made available, whether or not the customer uses it. Amounts billed or prepaid ahead of availability sit in deferred revenue; amounts earned ahead of billing sit in unbilled receivable.
- **On-demand and token usage** — recognised at the point of consumption, in the period metered. The simple case.
- **Milestone-delivered capacity** — recognised on acceptance, which is why `accepted_at` is a revenue field and not merely an operational one.

Two areas need judgement and therefore need to be explicit in the data model. Variable consideration — expected SLA credits and expected error credit-memos — is estimated and constrained at recognition; the `revenue_schedule` table carries it as a negative amount so the estimate is visible and auditable rather than implicit. And multi-element contracts bundling capacity, tokens, and support allocate on standalone selling price, with the allocated amount stored on the rated line.

The monthly tie-out (control C-12) is blocking: recognised revenue must equal invoiced revenue less the movement in deferred revenue plus unbilled receivable. A non-zero delta stops the GL post. Alongside it, the RPO

snapshot is taken daily and stored immutably, with the portion contingent on acceptance disclosed separately — so that any backlog figure quoted to a lender on any past date can be reproduced exactly, which is a different and much stronger claim than being able to compute today's.

12. Revenue Assurance

Telecom operators, who invented this discipline, consistently found one to three percent of revenue leaking before controls were in place. Usage-metered compute has the same anatomy — high event volumes, complex rating, negotiated exceptions — and no structural reason to do better. At the margins this business runs at, a point of recovered leakage carries no incremental cost of goods and is therefore worth roughly two points of EBITDA growth.

12.1 The three-way match

The backbone control compares what the control plane allocated (intent), what telemetry observed (reality), and what billing rated and invoiced. An intent-versus-reality break is an operational defect: capacity was allocated that did not run, or ran without being allocated. An intent-versus-billing break is a revenue defect. They have different owners and different urgency, and separating them is what keeps the queue actionable. Breaks above materiality open findings with an owner, a root-cause category, and a closure date, exactly as a trading firm runs a breaks process.

12.2 Leakage taxonomy

Mode	Mechanism	Detection
Unmetered usage	Agent crash, orphaned allocation, event loss	C-01 three-way match; M-08 orphan sweep
Meter-to-bill gap	Captured but dropped or late in aggregation	C-02 completeness on quantity and event count
Rating errors	Stale price book, expired discount still live, FX error	C-04 re-rate audit; C-11 four-eyes gate
Entitlement drift	Uncapped free tier, internal usage on production SKUs	C-05 margin-by-customer outliers
SLA over-crediting	Credits beyond the contractual formula	C-06 credit-memo to incident match
Fraud	Stolen cards, free-tier abuse, resale of discounted capacity	Velocity checks, KYC, anomaly detection
Contract leakage	Overage never billed, rollover mis-tracked, milestone missed	C-13 RPO conversion; K-01 drawdown checks
Tokenizer drift	A counting change silently re-prices every request	C-14 golden-request gate; drift view

12.3 KPIs

KPI	Definition	Target
Leakage rate	Identified leakage ÷ revenue, annualised	< 0.5% at maturity (1–3% typical at start)
Meter-to-cash yield	Cash collected ÷ metered value at contract rates	> 98%
Billing accuracy	1 – (billing-error credit memos ÷ billed revenue)	> 99.5%
Recovery rate	Leakage recovered ÷ identified	> 70%
Unbilled usage aging	Metered-but-uninvoiced value by age bucket	No bucket beyond one cycle

KPI	Definition	Target
Three-way match delta	Intent vs reality, capacity	< 10 bps daily
Token count delta	Gateway vs serving engine	0 tokens
IRN reporting	Documents unreported at T+21	0

13. Non-Functional Requirements

Dimension	Requirement	Why this number
Ingest throughput	1,000,000 events/sec sustained headroom	Phase-1 steady state is far below this; the headroom absorbs backfills and a 63k-GPU fleet reconnecting after a partition
Ingest latency	p99 ack < 50 ms; p99 emitted→queryable < 60 s	Producers must never block on billing; portal numbers must be same-minute credible
Metering completeness	≥ 99.99% of billable actions captured	A 0.01% loss budget on the merchant book is the practical floor of a distributed fleet
Rating determinism	Byte-identical re-rate, indefinitely	The evidence standard for disputes and statutory audit
Close cycle	Invoices issued within T+2 business days	Enterprise AP cut-offs; DSO
Billing ledger RPO / RTO	RPO 0 (synchronous replica), RTO 15 min	Losing an issued invoice is unrecoverable; losing an hour of analytics is not
Event store RPO / RTO	RPO < 5 min, RTO 4 h	Raw events are replayable from Kafka retention
Retention	25 months hot, 8 years cold, immutable	Companies Act s.128 books; GST records 72 months from annual-return due date
Availability	Ingest 99.99%; billing APIs 99.9%	Ingest loss is revenue loss; a billing API outage delays, not destroys

13.1 Security and data protection

The strongest privacy control in this design is architectural: the billing plane never receives prompt or response content. It receives counts, identifiers, and classifications. That boundary removes the worst-case breach surface entirely, and it should be enforced by schema — there is no field in the token payload that can carry content — rather than by policy.

Beyond that, events carry a `residency_zone` so that processing location is a queryable property rather than an assumption. Under the DPDP Rules notified in November 2025 and phasing to full enforcement in 2027, an organisation of Reliance's scale is very likely to be designated a Significant Data Fiduciary, which brings an India-based data protection officer reporting to the board, an independent data auditor, and data protection impact assessments for high-risk processing. Billing systems fall inside that audit scope, so the access model matters: read access to raw events is role-gated and logged, the `audit_log` table is insert-and-select only with no update or delete role in existence, and the e-invoice digital signature certificate is held in an HSM rather than in application configuration.

14. Delivery Plan

The sequencing principle is that metering correctness precedes billing sophistication. A platform that meters perfectly and invoices crudely can be fixed; one that invoices elegantly on bad meters cannot, because the errors are already in customers' hands and in the general ledger.

Phase	Scope	Exit criteria
P0 — Foundations (wks 1–8)	Event schemas frozen; ingest with idempotency, clock guard, quarantine; ClickHouse raw + rollups; producers instrumented on both legs	Three-way match runs daily and reports; no rating yet
P1 — Rate & invoice (wks 6–16)	Price book with four-eyes gate; rating engine with golden-vector CI; invoice assembly, numbering, PDF;	A full month rated in dry-run matches a manual model to the paise

Phase	Scope	Exit criteria
	INR domestic only	
P2 — India compliance (wks 12–22)	IRP/IRN integration; place-of-supply resolver; credit and debit notes; LUT gate; export and SEZ supply types	First live e-invoice registered; 30-day monitor green
P3 — Settlement & revenue (wks 18–30)	Payments, TDS application, FIRC matching, dunning, disputes; Ind AS 115 schedules; RPO snapshot	Revenue tie-out (C-12) zero for two consecutive months
P4 — Assurance & scale (wks 26–38)	Full control library automated; leakage workflow; margin-by-customer; multi-tenant, multi-currency; index-linked pricing	All BLOCKING controls have failed correctly in a game day

Two sequencing notes. India compliance (P2) overlaps P1 deliberately, because IRP integration has external dependencies — GSP selection, digital signature certificates, sandbox access — with lead times that are not under engineering control. And the assurance controls in P4 should be built incrementally from P0 rather than deferred: a control written after the system it audits is a control shaped by the system's bugs.

15. Artifact Inventory

The following ship with this document and are the normative contract. Where this prose and an artifact disagree, the artifact governs.

Path	Contents
schemas/usage_event.envelope.schema.json	The single event envelope; discriminated payloads; correction semantics
schemas/capacity_usage.payload.schema.json	GPU-hour metering; allocation state billing rules; acceptance gate
schemas/token_usage.payload.schema.json	Inference metering; disjoint cached tokens; tier and finish-reason rules
schemas/price_book.schema.json	Versioned pricing; flat, tiered, and index-linked models; four-eyes approval
schemas/rated_line_item.schema.json	The atomic unit of revenue; waterfall; input hash; tax and revrec classification
schemas/invoice.schema.json	Document model; GST/IRN block; FX; credit-note references
api/openapi.yaml	OpenAPI 3.1 — 18 operations across metering, pricing, rating, billing, settlement, revenue, assurance
ddl/postgres_billing.sql	Billing ledger; immutability triggers; four-eyes constraint; gapless numbering table
ddl/clickhouse_events.sql	Event store; per-leg rollups; quarantine; tokenizer-drift view
test/golden_vectors.json	17 conformance cases with computed expectations (regenerate via gen_vectors.js)
sql/reconciliation.sql	The assurance controls, executable
docs/CONFORMANCE.md	Control checklist with owners, cadence, and BLOCKING designations

16. Decisions Required Before Build

These are commercial, tax, and accounting decisions that engineering cannot make and should not default. Each one changes the schema, the tax logic, or the revenue treatment.

- **Legal entity and SEZ status** — which GSTINs sell what, and whether the Jamnagar campus operates as or supplies to an SEZ unit. This changes IRP supply-type codes, e-invoicing exemptions, and LUT handling, and it is the highest-impact open item.
- **Rollover policy** — the standard for take-or-pay contracts. Forfeit is simplest; rollover creates a tracked contract liability with an expiry and changes when revenue is recognised.
- **Depreciation life** — 4, 5, or 6 years for GB300. This does not affect billing but it drives the cost allocation behind margin-by-customer (control C-05), and an inconsistent assumption there makes the margin report unusable.
- **Provisioned token disclosure** — whether provisioned-throughput token traffic appears zero-rated on the invoice (recommended, for verifiability) or is omitted.
- **Index reference** — whether merchant physical contracts will reference the CME Silicon Data B200 index, the ICE/Ornn OCPI, or none. The index_linked price model is built for it; the choice is commercial and interacts with any treasury hedging programme.
- **DPDP designation** — confirmation of Significant Data Fiduciary designation and the resulting audit scope over billing systems, ahead of the 2027 enforcement date.
- **TDS profile defaults** — the expected withholding rate and section per customer class, to be loaded into account profiles so that cash application distinguishes withholding from dispute on day one.

References

- CBIC / GST Network: e-invoicing applicability, GST INV-01 schema, IRP registration flow, and the 30-day reporting limit; section 16 IGST Act (zero-rated supply); Form GST RFD-11 (LUT); section 170 CGST Act (rounding).
- Ministry of Electronics and IT: Digital Personal Data Protection Rules, notified 14 November 2025, with phased obligations for Significant Data Fiduciaries through 2027.
- Ind AS 115, Revenue from Contracts with Customers (converged with IFRS 15).
- CME Group: Silicon Data H100 and B200 Rental Index Futures, listed on NYMEX (launch targeted 5 October 2026); ICE and Ornn: cash-settled GPU compute futures on the Ornn Compute Price Index.
- Companion documents: "The Token and GPU-Capacity Business: An End-to-End P&L Primer" and "Compute Futures as a P&L Tool for Reliance Intelligence" (August 2026).

India tax and accounting treatments described here are engineering-facing summaries of publicly documented rules and must be confirmed by Reliance's tax and statutory audit teams before go-live. Thresholds and rates change; the platform reads them from configuration for exactly that reason.