SALESFUSION

Outcome Ledgers for Business AI

Technical architecture and implementation report

PUBLIC RELEASE · VERSION 1.0.0 · 2026-07-17

Outcome Ledgers for Business AI

Publication metadata

Field	Value
Author	SalesFusion Research
Publication date	2026-07-17
Version	1.0.0
Canonical slug	outcome-ledgers-for-business-ai
Canonical path	/technical-papers/outcome-ledgers-for-business-ai
Publication class	Technical architecture and implementation report
Approval state	Founder-approved public release

Revenue Cortex connection

Revenue Cortex needs outcome ledgers because growth systems are otherwise tempted to attribute pipeline, meetings, replies, and revenue to whichever agent produced the most confident narrative. Deterministic ledgers keep source records, contribution categories, reconciliation, and causal interpretation separate.

Evidence-status box

> SalesFusion uses an internal outcome-ledger and deterministic validation model for business AI reporting. The model preserves source records, metric definitions, effective time, contribution categories, outcome events, reconciliation checks, and learning records. This paper defines the architecture and describes deterministic reconciliation as an internal operating pattern. It does not make revenue, attribution, forecasting, decision-quality, or organizational-performance claims. Cross-system reconciliation and proxy-gaming evaluation remain required.

> The design uses append-only event history concepts and observability concepts consistent with the OpenTelemetry Specification. Risk framing is consistent with the NIST AI Risk Management Framework.

Abstract

Business AI systems are increasingly asked to report outcomes: pipeline movement, task completion, experiment results, operational progress, and contribution to objectives. If those reports are generated directly from prose, dashboards, or model summaries, the system can silently alter totals, conflate proxies with outcomes, erase null results, and overstate contribution. Outcome reporting needs a deterministic ledger first and AI explanation second.

This paper defines an outcome-ledger architecture for business AI. It is the outcome-layer companion to Paper 11: proof-of-run records verified execution, while the ledger records subsequent observations and limits causal interpretation. The ledger records source records, deterministic metrics, contribution categories, effective time, outcome events, reconciliation results, proxy risks, and learning records. AI may explain the ledger, propose interpretations, identify anomalies, or draft review notes. It may not invent events, overwrite metric definitions, backdate outcomes, silently change totals, or promote correlation to causation. SalesFusion uses an internal ledger and validator pattern; broader field claims require cross-system and proxy-gaming evaluation.

The problem

Business reporting is already difficult without AI. Source systems disagree. Timestamps differ. Records are duplicated. A task can be completed without contributing to an objective. A local metric can move favorably while a downstream guardrail worsens. A campaign can appear successful because the population changed. A team can count created artifacts rather than useful outcomes.

AI adds another risk: fluent explanation can make a weak ledger look rigorous. A model can summarize incomplete data, fill missing reasons, smooth over reconciliation errors, or imply that a workflow produced an outcome because the workflow happened before the outcome. That is not measurement. It is narrative over ambiguous records.

The problem is not that AI should be absent from reporting. The problem is that AI must be bounded by deterministic records. It should explain what the ledger contains and does not contain. It should not become the ledger.

Contributions

This paper contributes:

- A deterministic outcome-ledger model for business AI systems.
- Definitions for source records, metrics, effective time, outcome events, contribution categories, reconciliation, proxy risks, and learning records.
- A boundary between AI explanation and metric computation.
- Decision rules for preserving null, blocked, failed, negative, and indeterminate outcomes.
- An implementation pattern with validators and append-only events.
- An evaluation design for cross-system reconciliation and proxy-gaming resistance.

Core definitions

Source record is an immutable or versioned reference to the system, event, artifact, or human review that supports an outcome entry.

Deterministic metric is a metric computed by declared rules from source records. Its definition includes numerator, denominator, inclusion rules, exclusion rules, time window, effective-time semantics, deduplication logic, and owner.

Effective time is the business time at which an event applies, distinct from the time the system recorded it.

Outcome event is a dated record that something relevant to an objective occurred, did not occur, failed, was blocked, or was found indeterminate.

Contribution category labels the relationship between a system action and an outcome. Examples include direct execution evidence, assisted human action, prerequisite activity, correlated event, guardrail breach, no observed contribution, and unknown.

Reconciliation is the process of comparing ledger records, source records, external system totals, and metric definitions to detect missing, duplicated, stale, or inconsistent data.

AI explanation boundary is the rule that AI may summarize, question, and explain ledger records but may not create metric totals or alter outcome state without a validated event.

Proxy risk is the risk that a measured proxy can move favorably while the intended business outcome does not, or while a guardrail worsens.

Learning record is a bounded conclusion from reviewed evidence. It includes confidence, applicability limits, counterevidence, and whether a change proposal is warranted.

Ledger architecture

The outcome ledger is an append-only event system with deterministic projections. It is not a dashboard assembled from mutable notes. Each metric view is reconstructed from source records and metric definitions.

Layer	Function	Example objects
Source layer	Captures evidence references	source record, artifact, receipt, review
Event layer	Records business and workflow events	outcome event, blocker, failure, guardrail breach
Metric layer	Defines deterministic computation	metric definition, observation, window
Reconciliation layer	Checks totals and consistency	reconciliation run, discrepancy
Explanation layer	Produces bounded narrative	AI summary, anomaly question, review note
Learning layer	Records reviewed conclusions	learning record, change proposal

A minimal outcome event:

```
{
  "outcome_event_id": "out_001",
  "tenant_id": "tenant_ref",
  "objective_ref": "obj_retention_example",
  "metric_ref": "metric_verified_completion",
  "event_type": "verified_completion",
  "effective_at": "2026-07-16T00:00:00Z",
  "recorded_at": "2026-07-16T01:00:00Z",
  "source_refs": ["run_001", "artifact_001"],
  "contribution_category": "direct_execution_evidence",
  "value": 1,
  "confidence_label": "deterministic",
  "limitations": ["does_not_measure_business_value"]
}
```

The limitation field is important. A verified completion event may prove that a workflow produced an artifact. It does not prove that the artifact improved revenue, saved time, or changed a decision.

Metric definitions and contribution categories

Metric definitions must be explicit enough to be tested. A metric record includes:

```
{
  "metric_id": "metric_false_completion_rate",
  "name": "False completion rate",
  "numerator_rule": "runs_marked_complete_without_valid_artifact",
  "denominator_rule": "runs_marked_complete",
  "time_window": "calendar_week",
  "effective_time_field": "run_completed_at",
  "deduplication_key": "run_id",
  "exclusions": ["cancelled_before_start"],
  "owner_role": "operations_reviewer",
  "version": 3
}
```

Contribution categories provide controls against overclaiming:

Category	Meaning
direct_execution_evidence	The system directly performed the counted action and has receipt or artifact proof.
assisted_human_action	The system contributed a draft, analysis, or packet, but a human performed the action.
prerequisite_activity	The work prepared conditions but is not the final outcome.
correlated_event	The outcome occurred near the work but contribution is not established.
guardrail_breach	A protected metric worsened or policy was violated.
no_observed_contribution	Review found no supported contribution.
unknown	Evidence is insufficient to classify contribution.

This vocabulary lets reports say less when less is known. It also makes the mistake of treating every workflow output as an outcome visible and blockable.

Reconciliation and AI explanation boundaries

Reconciliation should run before explanation. It checks schema validity, source existence, duplicate keys, window boundaries, effective-time handling, metric-version consistency, and cross-system totals where available. A report with unresolved reconciliation errors should be labeled incomplete.

AI explanation can be useful after reconciliation:

- summarize which metrics moved;
- identify which records drove a change;
- surface missing data or anomalies;
- draft questions for review;
- compare outcome events with guardrails;
- explain contribution categories and limitations.

AI explanation must not:

- invent missing source records;
- change metric totals;
- silently drop null, failed, blocked, or adverse events;
- infer causality from sequence;
- rewrite contribution categories;
- promote a proxy improvement into business outcome success;
- approve a workflow or policy change.

The model can propose a learning record, but the learning record must be reviewed and bounded.

Implementation

SalesFusion uses an internal outcome-ledger and validator pattern. The implementation stores metric definitions, source references, outcome events, reconciliation runs, discrepancies, AI explanations, learning records, and change proposals as separate records. Deterministic validators run before a report is treated as current.

The core flow is:

```
capture source record or artifact
record outcome event with effective time
validate schema and source references
deduplicate by metric-specific key
compute deterministic metric projection
run reconciliation checks
label unresolved discrepancies
generate bounded explanation
create learning record only after review
create change proposal if evidence supports it
```

A learning record includes evidence, counterevidence, confidence, applicability limits, metric version, window, and reviewer decision. It cannot directly modify a workflow, policy, or metric definition. Any modification requires a change proposal, approval, versioning, and rollback plan.

Evaluation

The first evaluation should test ledger correctness, not business impact. A benchmark can provide synthetic source systems with known duplicates, late-arriving events, changed metric definitions, missing records, guardrail breaches, and proxy traps. The system should compute the expected totals and label unresolved discrepancies.

Metrics:

Metric	Definition
Reconciliation accuracy	Correctly identifies matching, missing, duplicated, and inconsistent records.
Metric projection accuracy	Deterministic totals match expected results for the declared definition.
Effective-time accuracy	Events are counted in the correct business window.
Null preservation	No-result, blocked, failed, and adverse states remain in reports.
Explanation faithfulness	AI summaries match ledger records and limitations.
Proxy-risk detection	Local metric gains with guardrail breaches are flagged.
Causal-overreach rate	Reports avoid unsupported causal language.

Business-outcome evaluation requires additional design: baseline, comparison group or other credible design, measurement window, sample, concurrent changes, confounders, and preserved null or negative results.

Risks and failure modes

The main failure is narrative overreach. A report can imply that AI produced an outcome because the AI system produced an artifact earlier. Another failure is proxy gaming: optimizing for easy-to-count activities such as completed tasks, messages drafted, or meetings scheduled while the actual objective does not move in the intended direction or a guardrail worsens.

Metric corruption is also common. Wrong joins, duplicate records, stale source snapshots, changed definitions, and inconsistent effective time can distort totals. AI explanation can hide these errors if reconciliation status is not visible.

Controls include deterministic metric definitions, append-only events, source references, reconciliation runs, guardrail metrics, contribution categories, learning-record limits, and review gates for change proposals.

Limitations

Outcome ledgers do not solve causal inference. They make records cleaner and claims more bounded, but they do not prove that a workflow caused a business result. They also depend on source-system quality and correct metric definitions. If the organization chooses poor proxies, the ledger can faithfully report the wrong thing.

SalesFusion's current evidence supports an internal ledger and deterministic reconciliation model. It does not support claims about cross-system accuracy at scale, resistance to gaming in the field, or business performance improvement.

Practical implications

Teams should separate measurement from explanation. Compute metrics deterministically, then let AI explain the computed records with visible limits. Reports should include reconciliation status, metric version, effective window, source coverage, contribution categories, and guardrail state. Null, blocked, failed, and adverse outcomes should remain visible.

Learning should be cautious. A completed experiment can create a learning record. It should not automatically change workflow policy, metric definitions, claims, or permissions.

Conclusion

Business AI reporting needs outcome ledgers before persuasive explanations. Deterministic metrics, source records, effective time, outcome events, contribution categories, reconciliation, proxy-risk controls, and learning records define checks intended to keep AI from inventing or silently altering totals. SalesFusion uses this internal ledger and validation pattern. The next evidence step is cross-system reconciliation and proxy-gaming evaluation, not stronger claims about business outcomes.

References

- OpenTelemetry, "OpenTelemetry Specification." https://opentelemetry.io/docs/specs/otel/
- National Institute of Standards and Technology, "AI Risk Management Framework." https://www.nist.gov/itl/ai-risk-management-framework

Changelog

- 2026-07-17 — Version 1.0.0: founder-approved first public release; publication metadata, evidence-status box, limitations, and verified references retained.