

SALESFUSION

The First Running Job

Technical architecture and implementation report

PUBLIC RELEASE · VERSION 1.0.0 · 2026-07-17

The First Running Job

Publication metadata

Field	Value
Author	SalesFusion Research
Publication date	2026-07-17
Version	1.0.0
Canonical slug	first-running-job
Canonical path	/technical-papers/first-running-job
Publication class	Technical architecture and implementation report
Approval state	Founder-approved public release

Revenue Cortex connection

A Revenue Cortex becomes credible when one real revenue job starts running with evidence before broader automation is attempted. The First Running Job is the activation wedge: a bounded, inspectable workflow such as a wake-up list, stale-deal radar, context packer, follow-up router, CRM truth loop, or approval console.

Evidence-status box

This paper describes an activation protocol SalesFusion uses and proposes an evaluation design. The protocol uses an Activation Card, non-mutating proof, evidence contract, approval boundary, live gate, rollback path, and expansion criteria. The allowed claim is protocol-level: a non-mutating, evidence-producing first job provides a more inspectable activation unit than launching broad autonomous scope at once. Any safety advantage remains a hypothesis until comparative activation and incident evidence is collected.

Abstract

The first live workflow in an agentic business system sets the operating pattern for everything that follows. If the first deployment is broad, mutating, weakly evidenced, and vaguely approved, later governance becomes cleanup. We define the First Running Job as a narrow activation unit that tests whether a workflow can use source-backed state, respect authority, produce expected artifacts, record run evidence, and pass a live gate before expansion. This paper specializes the broader Revenue OS installation sequence defined in Paper 06.

The protocol centers on an Activation Card: objective, scope, sources, authority, simulation plan, evidence contract, approval boundary, live criteria, rollback plan, and expansion criteria. The first job should run in simulation or non-mutating proof mode before any external consequence. Completion requires artifact and receipt verification, not process exit or agent self-report. SalesFusion uses the method as an activation protocol, and it still requires comparative evaluation before claims about lower incident rates or faster activation.

The problem

Agentic systems often fail at activation because the first workflow is chosen for excitement rather than inspectability. A broad workflow touches many sources, decisions, and external systems. When it fails, the team cannot tell whether the issue was memory quality, authority, connector permission, prompt behavior, evidence contract, or operational review. If the first workflow mutates external systems, errors can create cleanup work before governance is understood.

The more inspectable engineering move is to make the first job narrow and evidence-producing. The goal is not to establish that the whole system is valuable. The goal is to test the activation path: memory is source-backed, authority is explicit, simulation is possible, expected artifacts are defined, live approval is scoped, completion is verified, and rollback is known.

Contributions

This paper contributes the First Running Job protocol.

First, it defines the Activation Card as the pre-run contract for objective, scope, evidence, approval, live gate, rollback, and expansion.

Second, it specifies simulation and non-mutating proof as the default path before live activation.

Third, it defines an evidence contract for completion that rejects process exit or agent self-report as sufficient proof.

Fourth, it separates approval boundary, execution boundary, and expansion boundary.

Fifth, it proposes evaluation criteria for comparing activation approaches without claiming results.

Core definitions

First Running Job is the first narrow workflow activated in a governed business AI system. It is selected to test the operating path, not to maximize scope.

Activation Card is the structured artifact that must be complete before the job runs.

Simulation is a run against fixtures, dry-run adapters, or read-only data where no external mutation occurs.

Non-mutating proof is a real or realistic run that produces an artifact, recommendation, report, or verification output without changing an external business system.

Evidence contract is the predefined set of artifacts, receipts, logs, and checks required for `verified_complete`.

Approval boundary is the exact action, version, scope, actor, approver role, and expiry covered by approval.

Live gate is the decision point that allows the job to move from proof to bounded live operation.

Expansion criteria are the conditions required before adding more scope, actions, sources, or autonomy.

Activation Card

The Activation Card is the core object. It forces vague launch decisions into explicit fields by requiring the team to write the evidence contract before the run.

```
{
  "activation_card_id": "act_first_job_001",
  "job_name": "read_only_state_snapshot_job",
  "objective_refs": ["obj_activation_reliability"],
  "scope": {
    "tenant": "tenant_demo",
    "included_sources": ["src_current_policy_v1"],
    "excluded_actions": ["external_mutation"],
    "population": "bounded_fixture_or_approved_segment"
  },
  "memory_requirements": ["required_assertions", "freshness", "known_conflicts"],
  "authority": {
    "consequence_class": "read_only_or_internal_reversible",
    "approval_required_for_live": true,
    "approval_scope": "exact_workflow_version_and_action"
  },
  "simulation_plan": "run_with_non_mutating_adapter_and_expected_fixture",
  "evidence_contract": ["artifact_schema", "run_events", "trace", "blocker_state",
"verification_check"],
  "live_gate": ["proof_passed", "owner_reviewed", "rollback_tested"],
  "rollback_plan": "return_to_prior_version_or_disable_job",
  "expansion_criteria": ["zero_unauthorized_actions", "verified_completion", "operator_can_pause"]
}
```

The card should be short enough to review and precise enough to test. If a field is unknown, the job is not ready.

Protocol and state transitions

The First Running Job protocol has nine steps:

1. Select a narrow job with known objective, bounded scope, and known failure cases.
2. Build the Company Memory snapshot needed for the job.
3. Define the consequence class and authority boundary.
4. Write the evidence contract and expected artifact schema.
5. Run simulation or non-mutating proof.
6. Verify artifacts, events, blockers, traces, and restrictions.
7. Review the Activation Card and proof output.

8. Pass or fail the live gate.
9. If approved, run bounded live operation and monitor expansion criteria.

The run state machine is:

```
requested -> started -> output_created -> verification -> verified_complete
          |-> blocked -> resumed
          |-> failed -> retry | cancelled | rollback
```

The activation state machine is:

```
draft_card -> proof_ready -> proof_running -> proof_review
            -> live_gate_passed -> bounded_live -> expansion_review
            -> blocked | rejected | rollback
```

A blocked proof is a useful result. It can reveal missing sources, stale memory, wrong authority, connector drift, prompt injection, duplicate action risk, or an unrealistic artifact contract.

Evidence contract

The evidence contract defines what counts as completion. It should include:

Evidence	Purpose
input snapshot digest	identifies which memory state was used
workflow version	provides a check against approval/version mismatch
run events	reconstructs execution path
trace id	links infrastructure observation
expected artifact	shows useful output was produced
artifact validator	checks schema or content requirements
blocker record	explains incomplete or paused work
action receipt	records external action evidence, if any
verification result	separates output from verified completion
operator review state	records human decision

For a non-mutating proof, the action receipt may be a dry-run receipt or explicit no-mutation receipt. For a later live action, the receipt must be tied to the exact connector action and idempotency key.

Completion rule:

```

if process_exit == success and expected_artifact_missing:
    status = "verification_failed"
if artifact_valid and required_receipts_present and restrictions_respected:
    status = "verified_complete"
else:
    status = "blocked_or_failed"

```

This rule provides a deterministic check against false-completion claims. It also gives the team a concrete basis for debugging.

Approval boundary and live gate

Approval must be scoped to the exact workflow version and consequence class. A proof run may require only read-only authority. A live run that mutates an external system requires a separate approval. Approval includes action digest, version, scope, approver role, expiry, and conditions. Approval does not execute the action.

The live gate should require:

Gate item	Required evidence
memory	required assertions active, fresh, and sourced
conflicts	no unresolved required conflict
proof	non-mutating proof completed or blocked with accepted resolution
authority	approval valid for exact action and version
rollback	rollback or disable path tested
operator control	pause, inspect, reject, and export available
reporting	run evidence appears in reporting ledger

If any item fails, the correct output is not "try anyway." It is block, collect evidence, correct memory, revise the workflow, or escalate.

Rollback and expansion criteria

Rollback is part of activation, not an emergency add-on. For the first job, rollback may mean disabling the workflow, reverting to a prior workflow version, retracting a proposed memory change, or reconciling an external action if the live job was allowed to mutate. The rollback plan should be tested before broad scope.

Expansion criteria define what must be true before adding more workflows, broader populations, or more autonomy:

Criterion	Example
verified completion	proof or live runs meet evidence contract
no unauthorized action	authority suite records zero violations
no cross-tenant leakage	isolation tests pass
operator comprehension	operator can explain sources and approval path
blocker handling	known blockers are classified and reviewed
rollback proof	disable or version rollback works
reporting visibility	outcomes, costs, approvals, and failures visible

Expansion should be denied if the first job has unreviewed failures, unresolved authority ambiguity, missing rollback, or evidence gaps.

Implementation

SalesFusion uses the First Running Job as an activation protocol within the broader installation design. The implementation uses Company Memory, authority checks, workflow simulation, run evidence, reporting, and operator review.

Relevant interfaces:

```

POST /v1/state/snapshot
POST /v1/policies/evaluate
POST /v1/workflows/{workflow_id}/simulate
POST /v1/workflows/{workflow_id}/runs
POST /v1/runs/{run_id}/verify
POST /v1/runs/{run_id}/rollback
POST /v1/approvals
GET /v1/runs/{run_id}
GET /v1/audit/actions/{action_id}

```

Implementation invariants:

Invariant	Failure prevented
no broad first scope	unclear failure diagnosis
no mutation before proof	premature external consequence
no approval reuse	wrong-version execution
no self-reported completion	false done state
no hidden expansion	gradual authority creep
no untested rollback	unrecoverable activation

OpenTelemetry can provide trace and metric vocabulary; domain events provide business proof. RFC 9110 can support consistent HTTP interface semantics. Neither reference validates activation outcomes.

Evaluation

A proper evaluation would compare activation approaches. Possible conditions include broad immediate launch, narrow live launch, and First Running Job with non-mutating proof. Measures could include incident rate, unauthorized-action rate, false-completion rate, time to diagnose failure, rollback success, operator comprehension, and expansion decision quality. The study would need comparable settings, baseline, sample, time window, and disclosed limitations.

Before any field comparison, a scenario test suite should cover missing source, stale assertion, unresolved conflict, expired approval, wrong workflow version, prompt injection in source material, connector failure, duplicate retry, missing expected artifact, rollback unavailable, and attempted scope expansion.

Acceptance criteria for the protocol:

Criterion	Required behavior
incomplete card	job cannot start
proof failure	live gate blocks
artifact missing	run cannot be verified complete
approval mismatch	execution denied
duplicate retry	prior receipt returned or duplicate blocked
rollback unavailable	high-consequence live gate escalates
expansion request	checked against criteria before approval

These tests support protocol correctness. They do not prove safety advantage without comparative evidence.

Safety, risks, and failure modes

The First Running Job can fail if the job is narrow but irrelevant, non-mutating but unrealistic, approved by the wrong role, verified against a weak artifact, or expanded despite unresolved issues. Another risk is theater: teams may create an Activation Card but ignore it when launch pressure appears.

Controls include required card completeness, explicit exclusion lists, proof run evidence, authority binding, live gate review, rollback test, and expansion criteria. NIST AI RMF supports risk framing; OWASP LLM application threats are relevant to prompt injection and unsafe tool use; NIST SP 800-207 supports resource-scoped access reasoning. These references do not validate the protocol's incident effects.

Limitations

A First Running Job is not a full-system proof. It tests a narrow path. It may miss failures that appear only under larger volume, more connectors, more ambiguous objectives, or more consequential actions. A non-mutating proof can also understate live risk because real external systems behave differently from dry-run adapters.

The protocol also requires human discipline. If reviewers approve vague cards, skip rollback, or expand scope for convenience, the structure loses force. The method is best treated as a gate with audit consequences, not a checklist for appearance.

Practical implications

The first job should be selected for diagnosis value. A good candidate has bounded sources, clear expected output, known failure cases, non-mutating proof path, and visible objective relevance. A poor candidate requires broad authority, many connectors, unclear completion, or irreversible external consequences.

Operators should expect the first job to teach the installation what is missing. A blocked proof is not a failed project; it is evidence that the system found a boundary before live mutation. Expansion should follow verified operation, not the mere absence of visible errors.

References used

This paper uses NIST AI Risk Management Framework for risk framing, NIST SP 800-207 for resource-scoped access reasoning, OWASP Top 10 for Large Language Model Applications for threat awareness, OpenTelemetry Specification for observability concepts, and RFC 9110 for HTTP interface semantics. These references do not validate SalesFusion-specific activation outcomes.

Conclusion

The First Running Job is the smallest useful activation unit for a governed business AI system. It makes the launch question concrete: can this system use source-backed state, respect authority, produce evidence, pass review, and recover from failure on one bounded workflow? SalesFusion uses this activation protocol, but

comparative safety and incident claims still require evaluation. The practical rule is simple: test one narrow job with evidence before expanding autonomy.

Changelog

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