

GOVERNED TEMPLATE RENDERING REFERENCE: EVENT-DRIVEN ARCHITECTURE SPEC

DEFAULT V2 (DRAFT)

definition c0b8368e...d008

EVENT-DRIVEN ARCHITECTURE SPEC · PROJECT ARTIFACT

SPECIRA

Event-Driven Architecture: Dispatch Modernization

Meridian Field Services: instructional example, not project evidence

DRAFT · WATERMARK POLICY: DRAFT_ONLY

template event_driven_architecture v2 · pack: specira_default_delivery

the seam exists in the architecture; the event contract lives here

§1 Event Catalog

mandatory

1 decision

1 evidence rule

validators: every_event_has_producer_consumers_and_schema_version · every_topic_declares_its_compatibility_mode · every_event_carries_data_class_citations

The register: a schema registry alone is not governance; the catalog layers ownership and discovery. Provenance: the design walk against the container view, July 12 (building blocks^[1]).

Event · schema	Producer	Consumers	Classes
position.observed v2 backward · additive-only	Feed adapter	Board read model · position-store writer	RESTRICTED (<u>labels^[2]</u>) · payload cites <u>Position^[3]</u>
job.ready v1	Intake	Board · suggestion trigger	CONFIDENTIAL
assignment.created v1 via the outbox, §3	Dispatch API	Board · notification worker · audit writer	CONFIDENTIAL · cites <u>Assignment^[3]</u>
assignment.confirmed v1	Mobile confirmation	Board · audit writer	CONFIDENTIAL

Event · schema	Producer	Consumers	Classes
path			
override.recorded v1	Dispatch API	Audit writer · exception-review view · note- embedding hook (freshness ^[4])	CONFIDENTIAL worker-conduct
notification.requested v1	Assignment consumer	Notification worker → INT-2	CONFIDENTIAL

Every schema lives in the registry with its compatibility mode; a consumer addition is a registration, not a silent subscription; the embedding hook's addition to override.recorded was an explicit catalog change, dated.

§2 Event Topology

mandatory

1 decision

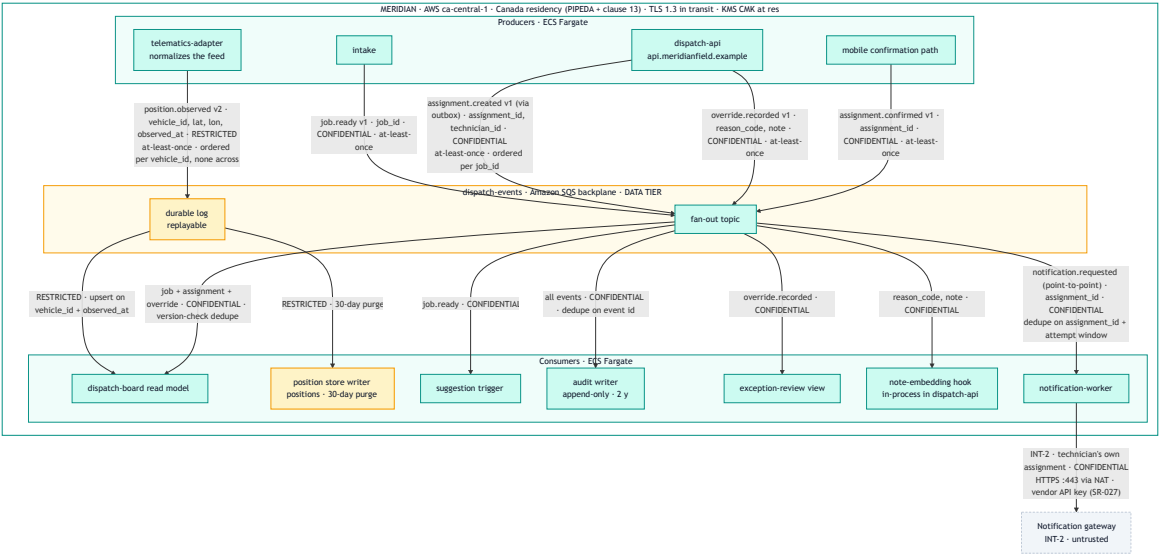
1 evidence rule

validators: every_partition_key_documents_its_ordering_scope · no_global_ordering_assumptions · topology_cites_architecture_seams · every_topology_edge_names_event_payload_fields_class_and_delivery_semantic · topology_nodes_name_deployed_service_citing_infrastructure_register

Ordering is a per-key promise; the forfeit is stated with the promise.

Shape, citing the seams (integration and data views^[1]): position.observed rides the durable log (replayable: the board can rebuild); assignment and override events ride the fan-out topic; notification.requested is point-to-point to the worker.

Key	Ordered (the promise)	NOT ordered (the forfeit)
vehicle_id	One vehicle's readings: the reconciliation the adapter's single-writer design needs (INT-1 design ^[1])	Across vehicles, and none is needed; the forfeit buys the fan-in parallelism the scale frame requires
job_id	One job's lifecycle: created before confirmed	Across jobs



Event topology: the fan-out under the specificity law. Fallback text: inside the Meridian boundary (AWS ca-central-1, Canada residency), producers publish to the dispatch-events SQS backplane: the telematics-adaptor emits position.observed (vehicle_id, lat, lon, observed_at · RESTRICTED · at-least-once · ordered per vehicle_id) to the durable log; intake, dispatch-api (via the outbox), and the mobile path emit job.ready, assignment.created (assignment_id, technician_id · CONFIDENTIAL · ordered per job_id), override.recorded, and assignment.confirmed to the fan-out topic. The durable log feeds the board (upsert on vehicle_id + observed_at) and the position-store writer (30-day purge); the fan-out topic feeds the board (version-check dedupe), the suggestion trigger, the audit writer (dedupe on event id · append-only), the exception-review view, and the note-embedding hook; notification.requested flows point-to-point to the notification-worker (dedupe on assignment_id + attempt window), which pushes INT-2 to the untrusted notification gateway (vendor API key · SR-027). Every edge names its event, fields, class, and delivery semantic.

§3 Delivery Semantics

mandatory 1 decision 1 evidence rule		
validators: every_delivery_semantic_choice_names_its_cost · every_consumer_declares_idempotency_strategy · no_unqualified_exactly_once_claims · every_dual_write_uses_the_outbox_pattern		
Topic	Semantic · cost	Idempotency strategy
position.observed	At-least-once inside the platform (receiver acks after enqueue); cost: duplicates on redelivery	Board upserts on vehicle_id + observed_at; a duplicate is a no-op
assignment / override topics	At-least-once; cost: duplicates	Audit writer dedupes on event id; the board

Topic	Semantic · cost	Idempotency strategy
		version-checks (<u>concurrency</u> ^[3])
notification.requested	At-least-once; a duplicate push is annoying, a dropped one breaks the confirmation flow; <i>the cost went the right way</i>	Worker dedupes on assignment id + attempt window

Exactly-once: claimed NOWHERE. The practical statement stands: at-least-once plus idempotent consumers. **Dual-write law:** assignment.created publishes via the OUTBOX: the Assignment row and the outbox row commit in one transaction; the relay publishes async (outbox rows^[3]); the gap this closes is the one the edge-case analysis names (concurrency^[5]).

§4 DLQ & Replay

mandatory

1 decision

1 evidence rule

validators: every_dlq_has_an_alerting_rule_and_drain_owner · replay_procedure_cites_idempotency_preconditions · realized_dlq_incidents_cite_issues

Every DLQ is owned and alerted through the observability design, never a second alert system.

Rule	Value
Retries	3, with backoff; then dead-letter with the failure class stamped (retriabl
Alert	The observability design's existing row: dead-letter rate above zero for 15 min, TICKET, p <u>catalog</u> ^[6])
Drain owner	Platform engineer; weekly floor even when quiet
Replay	Post-fix only; 10-event batch first; drain to zero. Safe because every consumer above dec position log tolerates full backfill (upsert), the notification path takes bounded replay only is user-facing

History: the truncated INT-3 file's partial load dead-lettered its batch (ISS-2^[7], RSK-08^[8]); the checksum gate that fixed it lives in the migration runbook (validation^[9]), and the drain that day followed this procedure.

§5 Open Questions

mandatory

1 decision

Question	Owner	Answer by	Blocks
Does the position log's retention (currently the 30-day purge window, retention rules ^[2]) also bound the board-rebuild window, or does the rebuild need a shorter operational guarantee stated separately?	E. Sandoval	Sep 30, 2026	The topology's replay note only

Refs

References & Package Contents

In the Specira workspace

specira	[1] Architecture: building blocks, integration/data views, INT-1 design app.specira.ai/projects/dispatch-modernization/artifacts/architecture
specira	[2] Data classification: labels, retention rules app.specira.ai/projects/dispatch-modernization/artifacts/data-classification
specira	[3] Data model: Position, Assignment, concurrency, outbox rows app.specira.ai/projects/dispatch-modernization/artifacts/data-model
specira	[4] Embedding & vector strategy: the freshness hook app.specira.ai/projects/dispatch-modernization/artifacts/embedding-vector-strategy#freshness
specira	[5] Edge case analysis: the dual-write concurrency rows app.specira.ai/projects/dispatch-modernization/artifacts/edge-case-analysis#concurrency
specira	[6] Observability: the dead-letter alert row app.specira.ai/projects/dispatch-modernization/artifacts/observability-monitoring#alerting
specira	[7] Issues: ISS-2 app.specira.ai/projects/dispatch-modernization/issues/iss-2
specira	[8] Risk register: RSK-08 realized app.specira.ai/projects/dispatch-modernization/artifacts/risk-register#rsk-08
specira	[9] Data migration: the checksum validation app.specira.ai/projects/dispatch-modernization/artifacts/data-migration#validation