GOVERNED TEMPLATE RENDERING REFERENCE: REAL-TIME & WEBSOCKET ARCHITECTURE DEFAULT V2 (DRAFT)

definition bc159736…7111

REAL-TIME ARCHITECTURE · PROJECT ARTIFACT

SPECIRA

Real-Time Architecture: Dispatch Modernization

Meridian Field Services: instructional example, not project evidence

DRAFT · WATERMARK POLICY: DRAFT_ONLY

template realtime_architecture v2 · pack: specira_default_delivery

the architecture commits the liveness target; this artifact owns the push mechanics

§1 Real-Time Channels

mandatory | 1 decision | 1 evidence rule

validators: every_channel_passes_the_real_time_worthiness_test · rejected_channels_recorded_with_polling_rationale · channel_payloads_cite_events_or_entities · every_channel_names_class_transport_port_auth_and_encryption

Push only when staleness costs something inside the poll interval, and a rejected channel is recorded like a choice.

Channel	Payload	Fan-out · frequency	Worthiness verdict
board.positions	position.observed events (catalog[1]) · RESTRICTED	Every connected dispatcher · ≤400/min (scale frame[2])	WORTHY : staleness costs assignment quality inside any reasonable poll interval; the 5-second target is the committed reason this channel exists (NFR-P1[3])
board.assignments	Assignment events · CONFIDENTIAL	Hub dispatchers + the ops-manager view	WORTHY : two dispatchers racing one job need the claim visible now (R-02 context[4])

Channel	Payload	Fan-out · frequency	Worthiness verdict
Technician mobile assignment list	*REJECTED to polling: the app polls on open and rides the INT-2 push notification for attention (INT-2[5]); a persistent connection to a field device buys battery drain and reconnect churn for updates the notification already delivers*		`rejected`
Suggestion results	*A request/response surface, not a channel; the worthiness test fails at the root*		`rejected`

Consumer contract: within-channel ordering per partition key only (topology[1]); at-least-once with client dedupe on event id.

Transport · auth · encryption (the specificity law on the register); both live channels stream over **server-sent events on HTTP/2 at :443**, **TLS 1.3**, authenticated by the dispatcher's OIDC session and narrowed by the **hub-scope filter before any write** (SR-001, permission model[8]); board.positions carries RESTRICTED positions, board.assignments CONFIDENTIAL assignment state (labels[11]). This artifact carries no diagram; the register is where the law binds, because the push path is a chain plus a one-to-many fan-out to identical browsers, which the labelled register conveys better than a box-and-arrow drawing would.

§2 Topology & Transport

mandatory · 1 decision · 1 evidence rule

validators: transport_losers_named_with_why · topology_names_its_backplane · push_path_segments_cite_event_catalog · push_path_edge_to_browser_names_transport_auth_and_scope_filter

Transport decision: server-sent events for both board channels
Why considered: the board is server-push-only (dispatcher actions travel the REST surface, endpoints[6]); SSE rides HTTP/2 through the existing gateway, needs no connection-upgrade operations, and carries a NATIVE resume token (the last-event id). **WebSocket rejected**: bidirectional capability the surface cannot use, at higher operational cost (the WebSocket-everything anti-pattern, named). **Long-polling**: the fallback tier, not the primary (realtime transport[7]).

Topology: API instances terminate SSE subscriptions statelessly; the queue is the backplane (the fan-out topic the events already ride: topology[1]), so any instance serves any dispatcher; no sticky sessions, no scaling wall, stated. **Push path, segmented**: adapter enqueue → topic fan-out → instance filter (hub scope: permission model[8]) → SSE write (position.observed RESTRICTED / assignment events CONFIDENTIAL, over TLS 1.3 to the dispatcher's own hub-scoped browser); the liveness budget's segments are named so a regression localizes, and the edge that leaves the platform names its class, transport, and scope so a reviewer reads the control off the path.

§3 Fallback & Degradation

mandatory · 1 decision · 1 evidence rule

validators: every_reconnect_strategy_uses_backoff_with_jitter · every_channel_has_a_missed_message_recovery_mechanism · full_resync_rule_stated_past_retention

Rule	Value
Reconnection	Exponential backoff from 1 s, ceiling 30 s, **FULL JITTER**: a hub-wide gateway restart must not re-arrive as a thundering herd
Disconnected state	The board marks positions stale, the same degradation posture the feed loss uses (quality scenarios[2])
Recovery	The SSE last-event id resumes the stream; the server replays from a 10-min buffer; past retention, **FULL RESYNC**: one read-model refetch (board read model[2]), then resubscribe live. The black hole between "reconnected" and "caught up" is closed by design (failure scenarios[4])
Polling fallback	If SSE is unavailable (a proxy stripping streams), the board polls the read model at 15 s with a visible degraded badge
Presence	*None: the board tracks vehicles, not dispatcher liveness; stated so the absence is a claim*

§4 Scaling Model

mandatory · 1 decision · 1 evidence rule

validators: scaling_budget_is_connections_times_rate_with_citation · horizontal_trigger_is_a_threshold · every_optimistic_update_declares_its_rollback_path

The budget multiplies connections by message rate, never a connection count alone. The honest statement is that pilot scale is trivial and the design is for hub-six.

Budget: 40 dispatcher connections at pilot × up to 440 events/min fanned per connection (positions plus assignment events at peak) ≈ 18 k SSE writes/min across 1 to 3 instances (scale frame[2], NFR-S1[3]), comfortably inside one instance's CPU-bound envelope. **Per-instance assumption**: active-messaging connections are CPU-bound, not memory-bound; the basis is that the message rate dominates at 11 writes/second per connection at peak. **Horizontal trigger**: 70% of the tested per-instance envelope adds an instance (scaling[9]); backplane fan-out cost grows linearly with instances and is the hub-six re-measure point (open questions[10]).

Reconciliation: the board is server-authoritative; the optimistic assignment claim renders pending and ROLLS BACK on stale_version, restoring the row and surfacing the notice (error registry[6], EC-03[4]).

§5 Open Questions

mandatory · 1 decision

Question	Owner	Answer by	Blocks
Does the 10-minute replay buffer hold at hub-six position rates, or does retention shrink with volume? (Buffer memory versus resync frequency; a measured answer.)	E. Sandoval	At the hub-six breakpoint re-run	The recovery table's retention row only

Refs References & Package Contents

In the Specira workspace

specira [1] Event-driven architecture: the catalog and topology the channels ride app.specira.ai/projects/dispatch-modernization/artifacts/event-driven-architecture

specira [2] Architecture: scale frame, quality scenarios, board read model app.specira.ai/projects/dispatch-modernization/artifacts/architecture

specira [3] NFR catalog: NFR-P1 liveness, NFR-S1 scale app.specira.ai/projects/dispatch-modernization/artifacts/nfr-catalog

specira [4] Edge case analysis: R-02, EC-03, the reconnect failure rows app.specira.ai/projects/dispatch-modernization/artifacts/edge-case-analysis

specira	[5] Integration inventory: INT-2 the notification path app.specira.ai/projects/dispatch-modernization/artifacts/integration-inventory#int-2
specira	[6] API contracts: the REST surface, the error registry app.specira.ai/projects/dispatch-modernization/artifacts/api-contracts
specira	[7] Decision log: the realtime-transport decision app.specira.ai/projects/dispatch-modernization/artifacts/decision-log
specira	[8] Auth & authz policy: the hub-scope filter app.specira.ai/projects/dispatch-modernization/artifacts/auth-authz-policy#permission
specira	[9] Infrastructure & deployment: the scaling policy app.specira.ai/projects/dispatch-modernization/artifacts/infrastructure-deployment#scaling
specira	[10] Performance test scenarios: the hub-six re-measure question app.specira.ai/projects/dispatch-modernization/artifacts/performance-test-scenarios
specira	[11] Data classification: the labels the channels carry app.specira.ai/projects/dispatch-modernization/artifacts/data-classification#labels