

GOVERNED TEMPLATE RENDERING REFERENCE: TEST STRATEGY DEFAULT V2 (DRAFT)

definition cd057622...fea5

TEST STRATEGY · PROJECT ARTIFACT

SPECIRA

Test Strategy: Dispatch Modernization

Meridian Field Services: instructional example, not project evidence

DRAFT · WATERMARK POLICY: DRAFT_ONLY

template test_strategy v2 · pack: specira_default_delivery

depth: standard · policy, never instances

§1 Test Shape & Approach

mandatory

2 decisions

1 evidence rule

validators: shape_decision_names_architecture_reason · manual_scope_is_deliberate_decision

The shape is a decision against this architecture: never a default ratio, never ASCII art. The size contract makes the layers enforceable in CI.

DECISION: SHAPE, PER CONTAINER

Pyramid for the dispatch API, a modular monolith ([ADR-1^{\[1\]}](#)): the logic lives inside one service, unit-heavy pays off. **Trophy** for the board, a typed component SPA: component-integration tests catch what unit mocks miss.

Layer	Target share	Rationale	Tooling
API unit	~70%	Monolith logic density	pytest
API integration	~25%	Queue + DB seams	pytest + local services
Board static + types	heavy	Trophy base	tsc · eslint
Board component-integration	dominant	Where board bugs live	vitest + testing-library
End-to-end	thin	Critical path only	playwright

Size contract: small (no network, under 60 s) · medium (local services only) · large (unrestricted, staging only). CI fails a mislabeled test. **Manual on purpose:** exploratory dispatch-floor sessions with M. Chen's team before cutover, the judgment calls a script cannot encode.

§2 Risk-Based Prioritization

mandatory

1 decision

1 evidence rule

validator: risk_claims_link_to_risk_register

Effort concentrates on	Risk entry	Suites
Assignment integrity	<u>double-assignment risk^[2]</u>	Race + idempotency suites
Feed-loss degradation	<u>telematics vendor risk^[2]</u>	Failure-injection suite
Propose-only rework containment	<u>union clarification^[2]</u>	Recommendation-module tests isolated behind its seam

Deliberately light: read-only reporting views, low blast radius, no contested logic; accepted by A. Reyes and E. Sandoval. Said explicitly, never implied.

§3 Coverage Targets

mandatory

1 decision

1 evidence rule

validator: coverage_targets_pair_number_tool_and_caveat

API: **85%** line floor, measured by the Python coverage tool. Board: **75%**, measured by the frontend runner. Excluded, listed: migrations, generated code.

THE CAVEAT, STATED AS LAW

Coverage proves execution, not correctness. Reviews grade assertion quality; a floor met with assertion-free tests is a finding, not a pass.

§4 Environments & Test Data

mandatory

1 decision

1 evidence rule

validator: test_data_approach_named_with_privacy_posture

Environment	Topology parity	Data	Integrations	Gates it may run
dev	Services only	Synthetic	Stubbed	Unit & integration
pilot	Production topology, Toronto scale	Synthetic + live telematics feed	Real	End-to-end & acceptance
production	Full (deployment view^[1])	Real	Real	Release verification only

Data class	Mechanism	Privacy posture
Technicians & jobs	Generated; edge-length names included (longest, empty, accented)	none
Positions	Generated tracks; live feed in pilot only	Location = personal-restricted (classification^[3]); never leaves production
Reason codes	The real governed list	Public/internal
Baseline comparisons	Production aggregates only (shadow window^[4])	No row-level copies

Quality metrics, reported monthly to the steering tier: defect escape rate (found-after-release / total) · critical-path suite duration · flaky quarantine count · mean time to fix a blocking gate.

§5 CI Gates & Verification Commands

mandatory

2 decisions

1 evidence rule

validators: gate_rows_carry_blocking_owner_and_timeline ·
command_palette_defined_once_and_runnable

Check	Layer	Blocking?	Owner	Fix timeline
Unit + types + lint + static security scan	static/unit	blocking on merge	feature team	pre-merge
Integration suite	integration	blocking	feature team	pre-merge
End-to-end critical path	e2e	informational daily → blocking at release	K. Yamada	2 business days

Check	Layer	Blocking?	Owner	Fix timeline
Dependency scan	supply chain	criticals block release (security gates ^[3])	eng lead	per severity SLA
Quarantined flaky tests	any	informational only	test author	30-day clock

Flaky policy: quarantine keeps the test running outside the required set, owner paged, thirty-day clock to fix or remove; never silent disablement. **Regression:** selective per change map; full suite nightly and pre-release. **Release gate:** automated for low-blast-radius; a human gate only for schema migrations, permission changes, and provider-gateway policy edits.

Command palette: defined once; stories reference it ([US-3 agent handoff cites these](#)^[5]):

```
api unit: pytest tests/unit -v api lint: ruff check app/ board suite: npm
run test e2e critical path: npm run e2e:critical types: npm run typecheck
```

§6 AI-Generated-Code Verification

conditional · engaged: stories ship agent-flagged

1 decision

1 evidence rule

validators: ai_deterministic_layer_uses_palette_commands · human_checkpoint_classes_named

Deterministic layer: every AI-authored change passes the palette's lint, type, static-analysis, and unit gates before any review opinion counts. **Human checkpoint:** required regardless of green gates for the named high-blast-radius classes (schema, permissions, gateway policy, the same classes as the release gate); the reviewer verifies boundary adherence (the story's always-ask-never list) and test meaningfulness, not style. **Story contract:** agent-flagged stories cite palette commands; the agent's checks and QA's checks are the same checks ([user stories](#)^[5]).

§7 Phase Gates & Defect Triage

mandatory

2 decisions

1 evidence rule

validators: phase_rows_carry_entry_and_exit_criteria · triage_cells_carry_timelines_with_distinct_owners

Phase	Entry criteria	Exit criteria
Unit & integration	Branch compiles, feature open	Coverage floors met · zero failing required checks

Phase	Entry criteria	Exit criteria
End-to-end on staging	Main green	Critical-path suite green · zero open criticals

Acceptance	48 stable staging hours	Per the story sign-off discipline (Acceptance & Sign-off^[5])
------------	-------------------------	---

Severity ↓ / Priority →	High priority	Low priority
Critical severity	Fix same day	Fix before next release
Minor severity	Fix this sprint	Backlog: no timeline pretended

Severity set by K. Yamada (technical impact); priority set jointly with A. Reyes (business urgency): two axes, two owners; conflating them is the classic triage failure.

§8 Acceptance Layers (Pointer)

mandatory · pointer only

validator: acceptance_pointer_links_resolve_zero_content

Nothing acceptance-shaped is written here; this section indexes where acceptance content lives, and that is its whole job.

Outlines: [the FRDs^{\[6\]}](#) (the only Given/When/Then anywhere). Concrete rows with test-case ids: [the stories^{\[5\]}](#). Executable definitions: [the test-case register^{\[7\]}](#); its exported view is the test-cases artifact. Example mapping gates Ready ([readiness gates^{\[5\]}](#)).

§9 Open Questions

mandatory

1 decision

Question	Owner	Answer by	Blocks
Does hub-six scale require a dedicated load test before rollout, engaging the performance-scenarios deliverable?	E. Sandoval	Aug 15, 2026	The release-gate posture for the rollout slice only

Omission note (how §6 renders on an all-human codebase): "§6 AI-Generated-Code Verification omitted: no agent-flagged stories, no AI-authored production code. Rationale recorded in adaptation event #1."

Refs References & Package Contents

In this export package

- package [4] Business Requirements Document: baseline shadow window ./brd-dispatch-modernization.docx
- package [6] Functional spec: scenario outlines AC-010..012 ./frd-recommendation-rules.docx

In the Specira workspace

- specira [1] Architecture: ADR-1, deployment view app.specira.ai/projects/dispatch-modernization/artifacts/architecture
- specira [2] Risk register: integrity, vendor, union entries app.specira.ai/projects/dispatch-modernization/risks
- specira [3] Security requirements: classification, testing gates app.specira.ai/projects/dispatch-modernization/artifacts/security-requirements
- specira [5] Story set: agent handoff, readiness gates, sign-off app.specira.ai/projects/dispatch-modernization/artifacts/user-stories
- specira [7] Test-case register: executable definitions app.specira.ai/projects/dispatch-modernization/test-cases