

Orchestra AI

AGENTS DRIVE · HUMANS NAVIGATE

The Reference Architecture

Orchestra Navigator™ · Patterns and Worked Examples

- The Framework defines what Governable Autonomy is.
- The Assessment Guide defines how it is measured.
- The Controls Catalogue defines how it is implemented.
- The Critical Controls Guide defines how strictly, by risk class.
- This document shows what all of it looks like, put together.

PREFACE

How to Read This Document

This is the only document in the family that does not introduce new rules. Everything here traces back to a control, a gate, or a determinant already defined in the four documents that precede it. Where this document differs is altitude: it shows patterns and worked examples rather than definitions. Most of this document is illustrative, and illustrations are inherently provisional until piloted against real engagements.

I

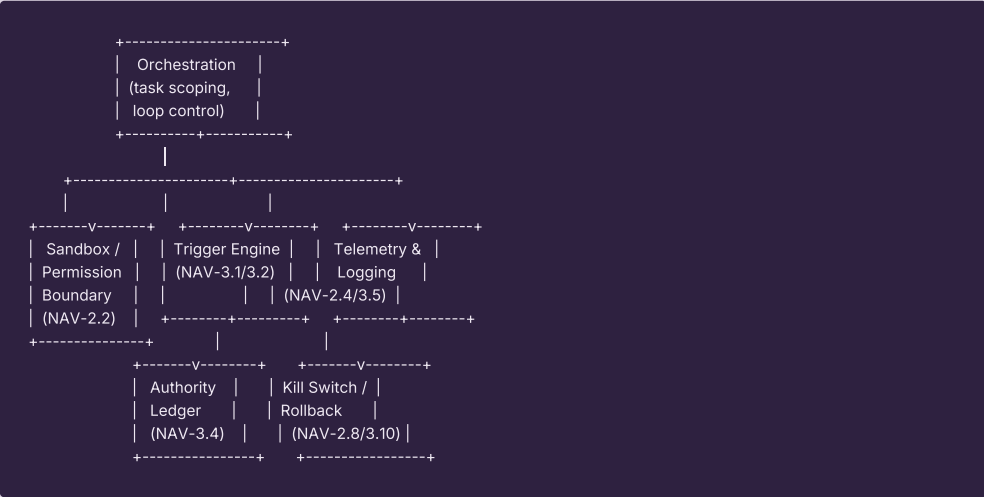
Purpose

Four documents now define Navigator's concepts, measurement method, controls, and risk-class stringency. None of them show what an actual harness looks like, what an actual engagement walkthrough feels like, or what the common ways teams get this wrong in practice actually are. This document exists to close that gap between definition and implementation.

II

Reference Harness Architecture

A harness capable of supporting Level 2 and above generally contains five components, regardless of the specific vendor or platform underneath it. This is a pattern, not a mandated architecture.



Orchestration scopes the task and enforces the loop's declared boundaries. **Sandbox/permission boundary** is the least-privilege execution environment. **Trigger engine** evaluates every action against the escalation catalogue. **Telemetry and logging** captures everything, including triggers evaluated but not fired. **Authority ledger** records every resolution. **Kill switch and rollback** provide the hard stop and recovery path. A harness missing any one of these cannot credibly claim Level 2.

III

Patterns by System Archetype

Pure code-generation or development loop

Typical harness: sandboxed repo access, test-suite gating, PR-based checkpoint. **Typical risk class:** Low to Moderate.

Common failure point: tests authored by the same agent that wrote the code, silently defeating the independence the checkpoint assumed. **Reference pattern:** require test authorship provenance distinct from implementation provenance.

RAG-based decisioning agent

Typical harness: retrieval pipeline plus execution sandbox plus trigger engine. **Typical risk class:** Moderate to High.

Common failure point: treating retrieval quality as solved because a vector database exists. **Reference pattern:** retrieval quality monitoring is not optional here; Knowledge Capability is Required, not Relevant, for this archetype.

Customer-facing conversational agent

Typical harness: conversation-scoped sandbox, escalation-to-human handoff, full logging. **Typical risk class:** Moderate.

Common failure point: a handoff path that exists in design but is too slow in practice, so customers route around it. **Reference pattern:** measure actual handoff latency as part of fatigue metrics.

Internal ops automation, low blast radius

Typical harness: lighter sandbox, checkpoint at deployment rather than every action. **Typical risk class:** Low.

Common failure point: scope creep into higher-stakes systems without reclassification. **Reference pattern:** review the loop-scoping document whenever permission grants change.

IV

Worked End-to-End Example

A hypothetical mid-sized fintech company, "Fictional Financial Co," walks a customer-facing loan-inquiry assistant from Level 1 to Level 3 over roughly eighteen months. Composed for illustration; it does not represent an actual client engagement.

MONTH 0, CHARTER

Three approvers classify the system: effortful reversibility, aggregate accountability, legal explainability, limited-risk regulatory classification, team/product blast radius. **Result: Moderate risk class, maximum permitted level 3.**

MONTHS 0–4, LEVEL 1

Every AI-drafted response reviewed by a support agent before sending. Provenance tagging distinguishes AI-drafted from human-drafted responses. Reviewer training completed.

MONTHS 4–9, LEVEL 2

Harness built: sandboxed access to approved templates and knowledge base only, enforced checkpoint before specific loan-term responses, full logging, demonstrated kill switch. Named harness owner assigned. Gate 1 to 2 evidence passes.

MONTHS 9–15, LEVEL 2 OPERATING HISTORY

The organization deliberately holds at Level 2 for two full quarters, using checkpoint-review findings to design the Level 3 escalation trigger catalogue, rather than guessing at triggers in advance.

MONTHS 15–18, GATE 2 TO 3

Trigger catalogue built from Level 2 findings. Adversarial evasion testing finds one gap, remediated before certification. Authority ledger and traceability verified. **System certified at Level 3.**

Eighteen months from Level 1 to a defensible Level 3 certification, with a deliberate two-quarter hold at Level 2 specifically to generate the evidence base the Level 3 controls needed.

An engagement promising Level 3 in weeks, for a Moderate-risk customer-facing system, should be treated with the same skepticism this guide asks assessors to apply to Critical-class systems roadmapping toward Level 3 they'll never reach.

V

Common Implementation Pitfalls Across Engagements

- **Building the harness before holding the operating history.** Skipping the Level 2 evidence-gathering period and designing Level 3 triggers speculatively. The single most common shortcut worth watching for.
- **Treating a vendor's certification as covering everything.** The assurance boundary gets skipped, and a vendor's SOC 2 report is assumed to cover controls it was never designed to address.
- **Confusing archetype defaults with a ceiling on ambition.** Knowledge Capability marked Not Applicable for a code-gen loop does not mean the organization can never build knowledge-grounded systems.
- **Letting the governance function exist on paper only.** Satisfied by an org chart entry rather than an actual named person who actually reviews things.
- **Over-classifying by instinct rather than by determinant.** "Customer-facing" or "involves AI" are not, by themselves, risk-class determinants.

VI

What This Document Does Not Yet Resolve

- Every pattern and worked example is illustrative, not drawn from an actual completed engagement. This document should be the first revised with real material once pilots proceed.
- The reference harness architecture is one reasonable shape, not the only one.
- As throughout the family, nothing here resolves the silent-escalation-failure problem.

The Navigator Family, Complete

This is the last document in the Navigator family's core five. If a question remains unanswered here, it is answered by one of its companions:

What is Governable Autonomy? The Framework whitepaper.

How is it measured? The Assessment Guide.

How is it implemented? The Controls Catalogue.

How strictly, by risk class? The Critical Controls Guide.

What does it look like, put together? This document.

**This document shows
what Governable Autonomy
looks like, put together.
The family is complete.
The pilots come next.**

Rajesh Srinivasan

Orchestra AI · rajesh@orquestraai.io
orquestraai.io

Orchestra Navigator™ Reference Architecture · Working draft, not yet released.

Trademark application in preparation, not yet registered.

© 2026 Orchestra AI.