

## 7. Technology Roadmap

This roadmap defines the technical sequence through which Field-Native demonstrates integrity under increasing load, agency, and expressive freedom. The system is required to prove stability before capability, and coherence before fluency. Each phase introduces controlled pressure and produces observable integrity evidence.

The arc progresses from executor-level constraints and metrics, through agent-mediated reasoning, to a field-native language model operating in raw-fidelity conditions with verifiable absence of drift or collapse.

### 7.1 Executor layer — integrity under action

The roadmap begins at the point of action, not generation.

This phase introduces constrained executors that operate within explicit meaning, authority, and epistemic boundaries. Executors are permitted to act only where constraints are satisfied and must refuse execution otherwise.

This phase establishes:

- Action primitives bound to explicit meaning objects
- Hard authority limits on what actions may occur
- Mandatory refusal modes when constraints are unmet
- Deterministic behaviour under identical conditions

**Integrity evidence produced:**

- Observable refusal rather than improvisation
- Repeatable execution without semantic drift
- Clear separation between “can act” and “cannot act”

No agents or language models are introduced at this stage.

### 7.2 Integrity metrics and telemetry

Once executors are stable, integrity is made measurable.

This phase introduces system-level metrics that track:

- meaning stability across executions
- authority boundary adherence
- epistemic status preservation
- contradiction accumulation vs resolution
- novelty containment

These metrics are structural, not behavioural. They do not measure success, speed, or output quality.

**Integrity evidence produced:**

- Quantifiable coherence under repeated execution
- Early detection of drift before visible failure
- Baseline integrity scores used downstream

These metrics become non-negotiable gates for all subsequent phases.

### **7.3 Agent layer — constrained reasoning under load**

Agents are introduced only after executor integrity is demonstrable.

Agents operate as reasoning coordinators, not autonomous actors. They may propose actions, generate hypotheses, and explore novelty, but cannot execute or promote conclusions beyond their authority.

This phase introduces:

- Agent reasoning bounded by executor constraints
- Explicit separation between proposal and action
- Multi-agent disagreement without forced convergence
- Preservation of unresolved ambiguity

**Integrity evidence produced:**

- Agents that cannot override executors
- Novelty surfaced as hypotheses, not decisions
- Stable behaviour under conflicting signals

Agents here function as *stress tests* for integrity, not optimisers.

### **7.4 Longitudinal continuity and drift detection**

The system is then required to remain coherent across time.

This phase introduces:

- Structured memory tied to meaning, not narration
- Longitudinal identity preservation
- Drift detection based on integrity metrics rather than output variance
- Accumulation of unresolved questions without collapse

Executors and agents operate repeatedly across extended time horizons.

**Integrity evidence produced:**

- No semantic drift across sessions
- No authority creep over time
- Stable handling of accumulated ambiguity

Only when time no longer degrades coherence does the roadmap proceed.

**7.5 Causal representation and intervention testing**

Causal structure is introduced as a constraint on reasoning.

This phase includes:

- Representation of causal hypotheses as provisional objects
- Support for interventions and counterfactual reasoning
- Invariant checks that reject spurious causality
- Requirement that causal claims survive executor-level validation

**Integrity evidence produced:**

- Causal claims that fail visibly rather than silently persisting
- Clear distinction between correlation and intervention
- Preservation of uncertainty where causality is weak

Causality here restricts novelty rather than amplifying it.

**7.6 Proof-carrying outputs and audit surface**

Before any expressive model is allowed, the system must explain itself structurally.

This phase introduces:

- Proof-carrying outputs for any consequential claim
- Full provenance chains from signal → meaning → inference → proposal
- Audit surfaces accessible without trusting the operator
- Integrity gates that block unprovable outputs

**Integrity evidence produced:**

- Outputs that include their own justification
- Inability to suppress uncertainty or contradiction
- Third-party verifiability of system behaviour

This phase makes integrity externally legible.

**7.7 Language model integration — expression without drift**

Only after integrity is demonstrable at executor, agent, causal, and audit layers is a language model introduced.

The LLM is treated strictly as an expressive surface.

This phase includes:

- LLM generation constrained by meaning, epistemics, and causality
- Mandatory attachment of epistemic status to all outputs
- Inability to speak beyond authority or evidence
- Suppression of fluency where integrity is uncertain

**Integrity evidence produced:**

- No hallucination under pressure
- No semantic drift across long interactions
- No collapse into persuasion or narrative smoothing

At this stage, the system demonstrates **raw-fidelity expression**: language that is accurate, bounded, and structurally honest.

## **7.8 Product integration as integrity witness**

Hayokai wearables and smartphone applications are introduced as real-world integrity stressors.

They provide:

- Longitudinal embodied signal
- User-owned context and ontology
- Real-world intervention/outcome loops

Product features are gated by integrity metrics, not demand.

**Integrity evidence produced:**

- Stable behaviour in lived environments
- No collapse under continuous input
- Preservation of user jurisdiction and meaning

Product here witnesses and invites integrity in human beings rather than driving it.

## **7.9 Conditional optimisation and scale**

Only once integrity has been demonstrated end-to-end is optimisation introduced.

This phase includes:

- Performance improvements within fixed semantic constraints
- Scaling infrastructure without altering meaning
- Load testing specifically for drift and collapse

**Integrity evidence produced:**

- Coherence preserved under sustained scale
- No degradation of epistemic boundaries
- Reversibility of scale decisions

## **7.10 Demonstration threshold**

The roadmap culminates in staged demonstrations:

- Executors refuse correctly
- Agents reason without override
- Integrity metrics remain stable
- Causal claims survive intervention
- Language remains raw, precise, and bounded
- No drift is observable across time, load, or novelty

At this point, the system will not need to ask to be trusted as it will irrefutably show that trust is structurally unnecessary.