

The Domain-Grounded Guardrail Build

An opinionated 14-day architectural build to eliminate dangerous regulatory drift by anchoring your AI coding agents in a deterministic understanding of your regulatory constraints.

THE EXECUTIVE DILEMMA

Why this? (The Functional Safety Trap): Large Language Models excel at generating code quickly, but they operate entirely on statistical probabilities. They do not know your unwritten physical invariants or strict regulatory constraints. When unleashed on a clinical, cyber-physical or mission-critical codebase with only natural language requirements, development agents confidently guess. They invent parallel state transitions, creating rogue domain logic that violates core safety architectures. This pushes a massive cognitive load downstream, taxing your most senior Systems Engineers and Functional Safety Leads with untangling hallucinated logic during code review.

Why now? (The Compliance Illusion): Transitioning to an AI-Assisted software development process introduces powerful implementation velocity. However, safety auditors will never accept a probabilistic LLM prompt as a safety guarantee. Natural language is ambiguous; dumping more documentation into the context window just confuses the AI. If engineering teams scale these tools without mathematically governed, machine-enforceable boundaries, the resulting regulatory drift introduces unacceptable compliance risks and existential liability.

The Shift-Left Solution: I engineer deterministic guardrails for development agents that systemically prevent AI-generated safety violations before they bottleneck your pull requests. We extract the tribal knowledge locked in your senior engineers' heads and codify it into a formal domain model and executable verification hooks. This gives your AI an executable vocabulary that instantly triggers local verification checks, forcing agents to self-correct against physical and regulatory reality before a single line of code is written.

01

THE DOMAIN-GROUNDED GUARDRAIL BUILD (14-DAY TURNAROUND)

Extracting core domain invariants, establishing enforceable boundaries, and embedding domain-specific guardrails into your AI-assisted SDLC.

To protect your agentic development workflows from regulatory drift, we engineer a localized, formal, and executable single source of truth for your core domain invariants. We do not boil the ocean. Instead, we isolate a single, high-value slice, translating fragmented business logic into a strict domain baseline. I then build the execution hooks, guides, and sensors to wire this boundary directly into the center of your agentic development framework.

This is a capability transfer, not a consulting retainer. At the end of this 14-day build, your engineering team owns a production-ready domain guardrail and the exact methodology required to scale this architecture internally across the rest of your platform, and across your organization, without ongoing vendor dependency.

ENTERPRISE OUTCOMES

- **Audit-Ready AI Compliance:** By translating physical invariants, clinical rules, ePHI access states or regulatory rules into an executable domain boundary, you generate deterministic proof that your AI cannot violate system safety. You can safely scale agentic coding workflows while generating the exact compliance artifacts auditors require.
- **Intercepting Catastrophic Trajectories:** By establishing the formal domain model as your executable single source of truth, conflicting specifications or hallucinated state transitions trigger immediate failures and feedback during agentic development. This blocks the agent from drafting a non-compliant pull request.
- **Protected Systems Engineering Bandwidth:** By codifying tribal knowledge into an executable baseline, your developers and autonomous agents are forced to operate using the exact same strict vocabulary. Instead of senior engineers burning hours acting as manual AI safety-checkers, the drop-in agentic hooks enforce your regulatory boundaries upfront, protecting your team's cognitive load.
- **Optimized Unit Economics & Latency:** By providing a machine-enforced formal contract, coding models no longer waste massive context windows trying to deduce your system physics. This drastically reduces token burn and rework, and safely routes execution to smaller, highly efficient models.

CORE DELIVERABLES

- **The Formal Domain Model & Invariant Baseline:** A formal, machine-readable domain model defining a single, high-value domain slice (a Bounded Context) and its associated invariants. This domain-specific model strictly maps the exact business rules, state transitions, and regulatory constraints your agents must respect to safely navigate that specific area of your business.
- **The Drop-In Agentic Guardrail:** I don't install heavy proprietary software. I deliver the exact, drop-in configuration files (agentic skills, computational guides, and hooks) that force your existing AI tools to comply with the domain model and force agentic self-correction. Operating quietly in the background, this AI-native configuration acts as a local pre-flight check, grounding agentic development actions in your regulatory reality without adding a single click to your developers' workflow.
- **The Architectural Evolution Runbook:** You are not trading AI babysitting for heavy DSL maintenance. To ensure your team retains 100% independent authority, I deliver a definitive maintenance runbook for your senior engineers. It provides the step-by-step workflow for your team to easily update, recompile, and extend the domain boundaries as your business rules evolve.

DELIVERY MECHANICS

- **Targeted Domain Extraction & Knowledge Source Mapping (90-Minute Working Session):** We conduct a single, high-efficiency working session with your cross-functional engineering leads, the only synchronous meeting on your calendar during the 14-day build. Together, we identify where it is highest-leverage to harden your agentic harness (isolating fragile state transitions, high-risk regulatory invariants, or frequent PR review failures) and pinpoint exactly where the ground-truth business rules already exist across your systems (existing OpenAPI contracts, database schemas, migration histories, or reference code). Your team does zero prep work or documentation authoring; I lead the session and extract the necessary pointers to conduct the domain mapping independently.
- **Rapid Domain Modeling & Guardrail Engineering:** Utilizing the knowledge sources identified in the kickoff, I independently extract and formalize your business rules into a formal domain model and regulatory invariants. I configure the customized agentic workflow setup, tailoring the automated verification hooks to your existing agentic development environment.
- **Air-Gapped Delivery & Zero-Access Integration:** I don't require access to your live CI/CD pipelines, internal networks, or sensitive source code. The compiled domain model and guardrail hooks are delivered as self-contained artifacts. This entirely air-gapped approach bypasses traditional InfoSec bottlenecks and vendor risk assessments, allowing instant activation.
- **Interactive Handover & Automated Verification:** During the Day 14 live handover, we run self-contained validation test fixtures inside your local environment, demonstrating the guardrail actively intercepting hallucinated logic and proving determinism across real agent workflows.

0 2

FINANCIAL STRUCTURE & TALENT ROI

The Domain-Grounded Guardrail Build

- *Project Timeline & Bandwidth:* Expected Turnaround: 14 Days. Internal Bandwidth Required: ~2.5 to 3 hours total across the two-week period (a single 90-minute live session to triage hardening targets and map domain sources, followed by two lightweight 30-minute asynchronous validation checkpoints with your lead engineer).
- *Investment:* 19 500 € (Excl. VAT). Structured as an **Operational Velocity Investment** sized specifically to fall within standard VP of Engineering / CTO discretionary spending limits, enabling immediate authorization without procurement committee delays.

The Talent ROI Justification:

This engagement directly buys back your most expensive and constrained asset: Senior Engineering Bandwidth.

When AI coding agents hallucinate business logic, the burden of untangling this hallucinated logic falls entirely on your senior developers during code review. If a €150k+/year senior engineer spends just 5 hours a week acting as an "AI spell-checker", they are wasting over €18,750 of capacity annually. By intercepting these errors deterministically upstream, this fixed-scope build systematically eliminates the PR bottleneck, paying for itself entirely within weeks through reclaimed senior velocity and protected DevEx.

0 3

NEXT STEPS

1. **Alignment Call:** A 30-minute introductory alignment to evaluate your current agentic SDLC bottlenecks, identify where your agents are hallucinating domain logic, and map out how this domain logic enforcement capability will integrate natively into your existing workflows.
2. **Targeted SOW Authorization:** A streamlined, fixed-scope Statement of Work is issued exclusively for the 14-Day Build.
3. **Project Kickoff & Build Launch:** Delivery launches on the agreed start date with the 60-minute session to pinpoint harness vulnerabilities, map domain sources, and deploy deterministic architectural safety within two weeks.

INTELLECTUAL PROPERTY STANCE

Arvoan operates on a clear "Outputs vs. Engine" IP model. The client retains 100% ownership of all customized deliverables (the compiled domain model and invariant baseline, the drop-in guardrail configurations, and agentic execution hooks). Arvoan retains exclusive ownership of its Background IP (the compilation scripts and methodology used to generate the schemas, *not* proprietary runtime binaries running in your environment), granting the client a perpetual license to utilize these embedded tools for internal operations.

OPERATIONAL ASSUMPTIONS

Maintaining the 14-day turnaround depends on timely feedback from designated technical stakeholders. To ensure continuous progress, intermediate asynchronous schema reviews operate on a standard 48-hour SLA. In the event that internal operational priorities require a review pause, the 14-day delivery clock halts day-for-day, safeguarding architectural thoroughness without cutting corners.

ABOUT TONI RUOKOLAINEN, PH.D.

I am a Principal Architect and Product Strategist specializing in Domain-Specific Modelling, Formal Methods and AI Enablement. I hold a Ph.D. in Computer Science with a focus on model-driven engineering, complex system modeling and formal methods, backed by over 15 years of deep technology research. Beyond academic rigor, I bring over a decade of hands-on industrial pragmatism, having served as a Founder, Product Manager, and Solution Architect designing complex enterprise systems. Because I move fluently between high-level architectural strategy and deep formal methods, I help mission-critical SaaS and cyber-physical engineering organizations accelerate their product delivery with AI-assisted SDLC built on an uncompromising, deterministic domain grounding.