AI Pipeline Governance Handbook

Ali Toygar Abak

AI Pipeline Governance Handbook

Ali Toygar Abak

This book is available at https://leanpub.com/aipipelinegovernance

This version was published on 2026-04-25

Leanpub

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

Contents

About This Book

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Who This Book Is For

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

What This Book Is Not

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Resources

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Introduction

The AI industry has a governance gap–not in policy, but in engineering.

Regulators write frameworks. Ethics boards publish guidelines. But when your AI system produces a harmful response at 3 AM, the question is not "do we have a policy?" The question is: **can you explain how the system got there, and can you prevent it from happening again?**

Most production AI systems cannot answer either question. They wrap a thin application layer around an LLM API call, and the entire decision-making process lives inside a black box. There is no decision trace, no intermediate state to inspect, no gate that was evaluated and passed.

This handbook introduces **pipeline governance**–an engineering discipline for building AI systems where every decision is structured, auditable, and controllable. The architecture treats the LLM as a sensor (a device that produces measurements about language and meaning) rather than as an oracle that produces truth. This single architectural decision unlocks a fundamentally different system design.

The Phionyx runtime implements this approach with a 46-block canonical pipeline. The LLM runs at position 12. Before it executes, the input passes through a kill switch gate, input safety validation, intent classification, and context retrieval. After it executes, the output passes through trust evaluation, ethics gates, behavioral drift detection, and cryptographically signed audit logging.

This book walks through every layer of that architecture with production code. You will learn how to build gate blocks that halt unsafe pipeline execution, configure governance without removing safety code, create custom blocks that integrate with the audit trail, implement tamper-evident logging with hash chains and Ed25519 signatures, monitor pipeline health with canonical metrics, and deploy the pipeline across different contexts while preserving governance properties.

The patterns work. They come from a system with 2,571 passing tests and four patent-pending applications. But more importantly, they represent an engineering approach to AI governance that you can apply to your own systems, regardless of which LLM provider or framework you use.

Why Pipeline Governance Matters

The AI industry has a governance problem, and it is not the one most people think about. The conversation around AI governance fixates on policy frameworks, regulatory alignment, and ethics boards. Those are important. But they miss the engineering layer entirely. The question that matters for working AI engineers is this: **when your system makes a decision, can you explain how it got there?** And more importantly, **can you prevent it from getting there again if the outcome was wrong?**

Most production AI systems cannot answer either question. This chapter explains why, introduces the concept of pipeline governance as an engineering discipline, and establishes the architectural principles that the rest of this handbook builds upon.

The Wrapper Problem

The dominant pattern in AI application development today is what I call the "wrapper architecture." It looks like this:

```
User Input --> Prompt Template --> LLM API --> Parse Response --> Return to User
```

A thin application layer wraps an LLM call. The wrapper adds context, maybe retrieves some documents via RAG, formats a prompt, sends it to the model, and parses the response. The entire decision-making process lives inside the LLM call. The application code is a glorified HTTP client.

This pattern is fast to build. It produces impressive demos. And it is ungovernable.

When the LLM produces a harmful response, the application has no internal record of why. There is no decision trace, no intermediate state that can be inspected, no gate that was evaluated and passed. The response emerged from a black box, and all the application knows is that it sent a prompt and received text back.

Governance requires structure. You cannot audit what you cannot see. You cannot control what you cannot decompose. The wrapper pattern fails at governance not because the LLM is inherently unsafe, but because the application provides no structural surface for governance to attach to.

Why LLM-First Architectures Fail at Governance

The wrapper problem is a symptom of a deeper architectural mistake: treating the LLM as the decision-maker rather than as a sensor. When you design your system around the assumption that the LLM is the source of truth, several things follow:

No intermediate checkpoints. The decision is atomic–it happens in a single API call. There is nowhere to insert a safety check between "the model thought about the problem" and "the model produced a response."

No reproducibility. Given the same input, the LLM may produce different outputs. Without deterministic pipeline state, you cannot replay a decision to understand what happened.

No separation of concerns. Ethics checking, safety validation, input filtering, output formatting, and core reasoning all collapse into prompt engineering. When everything is a prompt, nothing is auditable independently.

No graceful degradation. When a wrapper application encounters an edge case, it either returns whatever the LLM says or crashes. There is no fallback logic, no degraded-but-safe operating mode, no circuit breaker.

Consider a concrete scenario: a user sends input that triggers a harmful response. In a wrapper architecture, your options are limited to prompt-level interventions–system instructions that say "don't be harmful." In a governed pipeline, the input hits a safety gate before it reaches the LLM. The gate evaluates the input against defined risk criteria. If the risk is elevated, the pipeline can reject the input, defer to a human reviewer, or route to a safer processing path. Every decision is logged with the specific risk scores that informed it.

Treating the LLM as a Sensor

Phionyx is built on eight binding axioms called the Echoism Core specification. The first and most important axiom states: **"LLM output is measurement, not**

truth.”

This is not a philosophical position. It is an architectural decision with concrete engineering consequences. When you treat the LLM as a sensor–a device that produces measurements about language and meaning–rather than as an oracle that produces truth, you unlock a fundamentally different architecture:

```
User Input --> [46-Block Deterministic Pipeline] --> Governed Response
                        |
                        |--> Block 1:  Kill Switch Gate (fail-closed)
                        |--> Block 3:  Input Safety Gate
                        |--> Block 4:  Intent Classification
                        |--> Block 12: Cognitive Layer (LLM call)
                        |--> Block 16: Trust Evaluation
                        |--> Block 17: Ethics Pre-Response
                        |--> Block 18: Deliberative Ethics Gate
                        |--> Block 24: Behavioral Drift Detection
                        |--> Block 37: Response Build
                        |--> Block 43: Audit Layer
                        ...
```

The LLM is one block in a 46-block pipeline. It runs at position 12. Before it executes, the input has already passed through a kill switch gate, an input safety gate, intent classification, and context retrieval. After it executes, its output passes through trust evaluation, ethics gates, behavioral drift detection, and audit logging before reaching the user.

This is pipeline governance: **a deterministic, ordered sequence of blocks where each block has explicit inputs, outputs, pass/fail criteria, and an audit trail.** The LLM is not removed from the system–it is essential for cognitive processing. But it is situated within a structure that can govern its behavior.

What Pipeline Governance Provides

Pipeline governance is not a buzzword. It is a set of concrete engineering properties:

Auditability

Every block execution produces a `BlockResult` with a status, data payload, and optional error. The audit layer block at position 43 computes integrity scores and links each turn into a hash-chained audit record signed with Ed25519. When something goes wrong, you have a complete trace of every decision that led to the outcome.

Fail-Closed Safety

The kill switch gate at position 1 evaluates four shutdown conditions every turn: catastrophic ethics violation (risk > 0.95), meta-cognitive trust collapse (T_meta < 0.1), sustained behavioral drift (5+ consecutive drift detections), and manual trigger. If the evaluation itself fails, the gate triggers shutdown—fail-closed by design. This is a property you cannot achieve in a wrapper architecture because there is no structural position for the gate.

Separation of Concerns

Ethics is not a prompt. It is a dedicated block (deliberative_ethics_gate at position 18) that evaluates actions against four ethical frameworks: deontological, consequentialist, virtue, and care. If three or more frameworks deny an action, the pipeline forces early exit. If two deny, the action defers to human review. These are deterministic rules evaluated on structured risk vectors, not LLM-generated text about ethics.

Determinism

The pipeline executes blocks in a canonical order defined by a versioned contract (v3.8.0, 46 blocks). Given identical inputs and identical block configurations, the non-LLM blocks produce identical outputs. The LLM block (cognitive_layer) is the only source of nondeterminism, and it is explicitly bounded—its output is treated as measurement and validated by downstream blocks.

Policy Without Code Changes

Gates are never removed from the pipeline—they are policy-bypassed. A `RunCapabilities` configuration determines which gates are active without modifying the pipeline structure. This means you can configure governance levels for different deployment contexts (development, staging, production) without touching the code that implements the governance logic.

The Cost of Not Governing

The case for pipeline governance is often made in terms of benefits. But the cost of not governing is more instructive.

Without pipeline governance, incident investigation reduces to reading LLM prompts and responses. There is no intermediate state to inspect, no block-level timing to identify bottlenecks, no gate evaluation log to explain why a harmful response was allowed through. When a regulator asks "why did your system produce this output?", your answer is "the LLM generated it"–which is not an answer.

Without pipeline governance, safety changes require prompt engineering. Every safety improvement is a modification to a system prompt that competes for token budget with the actual task. Safety and functionality are not separated; they are entangled in natural language instructions that the LLM may or may not follow.

Without pipeline governance, testing is limited to input-output pairs. You can test that certain inputs produce expected outputs, but you cannot test that internal safety gates evaluated correctly. You cannot write a negative test that proves a dangerous input was rejected by the ethics gate rather than accidentally avoided by prompt engineering.

Architecture Principles

The rest of this handbook builds on four principles that flow from the sensor-based approach:

Principle 1: Blocks, not prompts. Every governance concern gets its own pipeline block with explicit inputs, outputs, and pass/fail criteria. Ethics is a block. Safety is a block. Audit is a block. They are not prompt instructions.

Principle 2: Gates fail closed. A gate that cannot evaluate its conditions must trigger its protective behavior. The kill switch fails closed. The input safety gate fails open (because blocking input on an error is overly conservative), but this is an explicit, documented design decision for each gate.

Principle 3: Audit is structural. Audit is not log statements scattered through code. It is a dedicated pipeline block that computes integrity scores, builds hash chains, and produces cryptographically signed records. Audit cannot be bypassed because it is part of the canonical pipeline order.

Principle 4: Core does not know about delivery. The pipeline, blocks, and governance logic live in `phionyx_core/` and have no knowledge of HTTP, databases, or web frameworks. This separation means the governance logic can be tested independently, deployed in different contexts (web server, CLI, embedded), and reasoned about without considering delivery-layer concerns.

What This Handbook Covers

This handbook walks through the Phionyx pipeline governance architecture in detail, with code from the production implementation:

- **Chapter 2** explains the 46-block canonical pipeline, its ordering contract, and execution model.
- **Chapter 3** dives into gate architecture: input safety, ethics, and kill switch gates.
- **Chapter 4** covers capability profiles and how to configure governance without removing gates.
- **Chapter 5** shows how to build custom pipeline blocks that integrate with the governance framework.
- **Chapter 6** details the audit trail: hash chains, Ed25519 signatures, and tamper-evident logging.
- **Chapter 7** covers monitoring and telemetry: what to measure, how to alert, and what the metrics mean.
- **Chapter 8** addresses deployment patterns: the bridge layer, echo server integration, and production configuration.

Each chapter references real source code from the Phionyx implementation. The patterns are not theoretical–they are extracted from a working system with 46 active pipeline blocks, 2,571 tests, and a Composite Quality Score of 0.862 measured across 291 parameter experiments.

Key Takeaways

- The "wrapper architecture" (thin app around an LLM call) is inherently ungovernable because it provides no structural surface for governance controls.

- Treating the LLM as a sensor rather than an oracle enables a pipeline architecture where governance blocks surround the LLM call.
- Pipeline governance provides auditability, fail-closed safety, separation of concerns, determinism, and policy configuration without code changes.
- The Phionyx 46-block pipeline places the LLM at position 12, with safety gates before and after it, ethics evaluation, behavioral drift detection, and cryptographically signed audit records.
- Gates are never removed–they are policy-bypassed. This ensures that the governance structure is always present, even when specific gates are inactive.
- Core governance logic must be framework-agnostic (no HTTP, no database dependencies) to enable independent testing and multi-context deployment.

The 46-Block Canonical Pipeline

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

The Canonical Block Order

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Contract Versioning

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

The Execution Model

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

BlockContext: Shared State

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

BlockResult: Structured Outcomes

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

The Orchestrator

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Pipeline Execution Flow

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Block Dependencies

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Block Categories

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Boundary Gates (positions 1, 3, 12, 18, 22)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Cognitive Processing (positions 4-13)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Evaluation (positions 14-17, 23-24)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

State Management (positions 25-36)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Output and Learning (positions 37-46)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

The Block Factory

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Key Takeaways

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Gate Architecture

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

What Makes a Gate Different

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

The Kill Switch Gate

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Position and Purpose

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Implementation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Four Trigger Conditions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Fail-Closed Design

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

NaN Guard

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Kill Switch States

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

The Input Safety Gate

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Position and Purpose

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Two-Phase Evaluation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Fail-Open Design

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

The Deliberative Ethics Gate

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Position and Purpose

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Four-Framework Deliberation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Dependency Declaration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Gate Design Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Pattern 1: Structured Result Data

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Pattern 2: Protocol-Based Injection

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Pattern 3: Inline Fallback

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Key Takeaways

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Capability Profiles and Policy Bypass

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

The RunCapabilities Configuration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

The RBAC Permission Matrix

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

The RBACManager

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Dynamic Permission Management

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Configuring Policy Without Removing Code

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Level 1: Block Skip (coarsest)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Level 2: Gate Threshold Adjustment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Level 3: RBAC Matrix Customization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Level 4: Block Implementation Swap

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Anti-Patterns to Avoid

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Anti-Pattern 1: Conditional Block Registration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Anti-Pattern 2: Hardcoded Permission Bypass

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Anti-Pattern 3: Environment Variable Gates

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Key Takeaways

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Building Custom Pipeline Blocks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

The PipelineBlock Interface

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Building a Custom Gate Block

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Building a Custom Processing Block

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

The Factory Pattern

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Testing Custom Blocks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Block Implementation Checklist

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Key Takeaways

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Audit Trail Integration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

The AuditRecord Schema

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Hash Chain Integrity

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Computing the Record Hash

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Verifying Chain Integrity

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

The Genesis Record

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Ed25519 Digital Signatures

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Signature Workflow

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

The Audit Layer Block

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Inline Fallback

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Input and Output Hashing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Decision Trace for Explainability

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Audit Trail Verification

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Storage Considerations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Key Takeaways

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Monitoring and Telemetry

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

The Telemetry Contract

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

The TelemetryCollector Protocol

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Canonical Metrics

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Block Timing Metrics

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Gate Trigger Metrics

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Kill Switch State Metrics

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Execution Guard Metrics

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Audit Integrity Metrics

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

RBAC Metrics

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Block-Level Dashboards

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Alerting Strategy

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Critical (Immediate Page)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Warning (Investigate Within 1 Hour)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Info (Review Daily)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Correlating Telemetry with Audit Records

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Pipeline Health Score

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Key Takeaways

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Deployment Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

The Core/Bridge Separation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

The Bridge Layer

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Concrete Implementations for Core Protocols

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Dependency Injection Wiring

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

HTTP API Surface

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Production Configuration Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Pattern 1: Profile-Based Configuration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Pattern 2: State Persistence

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Pattern 3: Graceful Service Discovery

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Pattern 4: Rollback on Block Failure

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Pattern 5: Parallel Execution with Governance Constraints

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Deployment Checklist

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Pre-Deployment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Runtime

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Post-Deployment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Multi-Context Deployment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.

Key Takeaways

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/aipipelinegovernance.