COLDCODE SI LABS LIBRARY

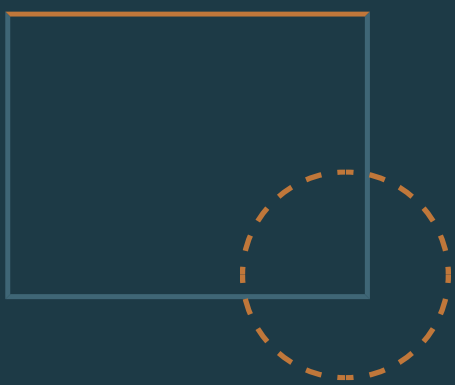
Proof of Mechanism

How leaders who build with AI agents tell a working product from a convincing demo

Theories, frameworks and one candid case for founders, CTOs and directors

Courage Garikai

Funded by Coldcode Si Labs



FIRST EDITION · 2026

PROOF OF MECHANISM

Copyright © 2026 Courage Garikai.

Published by Coldcode Si Labs Library, Harare, Zimbabwe.

First edition, 2026.

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted
form or by any means without the prior written permission of the publisher, except in the case
quotations in a review or certain other noncommercial uses permitted by copyright law. The fr
worksheets and checklists in this publication are meant to be used, copied and adapted by the

This is a work of analysis and practical guidance. It is not financial, legal, investment or profes
advice, and no outcome is promised: results depend on the reader's own market, effort and juc
Platform behaviour and figures were checked at the time of writing and change.

COLDCODE SI LABS LIBRARY · Harare, Zimbabwe

THE BOOK

Contents

About this edition 4

PART I · The new illusion

1 · The demo that wasn't 10
2 · Why surfaces got cheap 16
3 · Simulated completeness 22
4 · Trust products and credence goods 28

PART II · Anatomy of a convincing demo

5 · Fabricated state 34
6 · Theatrical actions 39
7 · Unenforced controls 45
8 · Dangling references 51
9 · Borrowed authority and test theatre 57

PART III · The mechanism audit

10 · The six probes 64
11 · The one-hour outsider audit 70
12 · Synthetic evidence and the third party 76
13 · Reading an audit as a leader 82
14 · Rebuild or repair 87

PART IV · Building for proof

15 · Mechanism first 94
16 · One source of truth for every claim 100
17 · Honest limits as a product feature 106
18 · Truthful outreach 112
19 · Status honesty 118
20 · Verifiability as strategy 124

PART V · Leading AI-built work

21 · Governing agents with evidence 131
22 · Metrics, incentives and Goodhart 137
23 · Law, ethics and reputation 142

24 · A ninety-day proof-of-mechanism programme 147

TOOLKIT · Frameworks, guide, glossary and references

25 · Framework compendium 154
26 · Facilitator's guide 162
27 · Glossary 167
28 · References 172

About this edition

First edition · 2026

This book is written for founders, chief technology officers, product leaders, directors and anyone who commissions, accepts or oversees software built with AI coding agents. It is a discussion book. Each chapter sets out a question, the theory that helps answer it, a working case, a framework you can use, the points on which reasonable people disagree, questions for discussion and a short exercise.

The case is AttestOS, a bot-verification service for websites. Its first version, committed to a public repository on 26 October 2026, looked finished and was almost entirely surface. An audit the next day found 65 problems, 18 of them critical. The service was rebuilt the same day and checked again live on 28 October. AttestOS belongs to the same owner who commissions the work of Coldcode Si Labs. The audit and the rebuild were carried out by the owner working with AI coding agents, and the same process produced the analysis in this book. It is therefore a bounded case, not an independent evaluation. At the time of writing AttestOS had **no customers and had taken no payments**. Nothing in this book is evidence that it works against real attackers or succeeds commercially.

This book does not say how the first version was produced. Some of its defects resemble failure modes documented in AI systems. They are equally consistent with a person building a demonstration in a hurry. Where that question arises it is treated as open.

Throughout, three kinds of statement are kept apart:

- **Research findings** are cited to their sources.
- **Facts about the case** come from its public commit history, the itemised audit report, the production database and its own usage meter, as of October 2026.
- **Proposed frameworks, constructed examples and recommendations** are labelled as such. They are proposals to test, not proven laws.

No professional biography, endorsement, ISBN or independent audit is claimed. The author and funding credits are those specified by the commissioning owner. AI tools assisted in drafting this book; the author is responsible for its content.

© 2026 Courage Garikai. All rights reserved. Published by Coldcode Si La Library, Harare. No part may be reproduced without written permission, the frameworks and exercises, which are meant to be copied.

The promise of this book

For most of the history of software, a product that looked finished was usually close to finished. Screens, dashboards and manuals were expensive, so they were built last, on top of machinery that already worked. Anyone shown a polished demonstration could reasonably assume there was something real underneath. Buyers, investors, boards and managers relied on that assumption every day without noticing it.

That assumption no longer holds. An AI coding agent can produce a landing page, an owner console full of charts, a customer portal, product covers, user manuals, a test suite and a confident release note in an afternoon. It can produce all of them whether or not anything behind them works. The *surface* of software has become cheap, while the *mechanism* (the stored data, the enforced rule, the message actually sent, the payment actually taken) still costs what it always cost to build and to check.

The result is a new kind of risk: products that are convincing and incomplete at the same time, without anyone necessarily intending it. This book calls that condition **simulated completeness**. It argues that leaders who build with AI need a new habit to match. They should stop accepting surfaces and start asking for *proof of mechanism*. The title borrows from the case. AttestOS sells "proof of work", a small computation that a visitor's browser must perform to show it is not a cheap automated script. This book asks leaders to demand the equivalent from every product they accept: evidence that the work behind each surface has been done.

Who this book is for

- **Founders and solo operators** who ship with AI agents and must decide what to trust in what the agents produce.

- **CTOs, engineering and product leaders** who accept work from agents, contractors and teams, and need acceptance criteria that cannot be satisfied by appearance alone.
- **Directors, investors and board members** who are shown products and asked to rely on them, and who need questions that get past the demonstration.
- **Buyers of software, especially of security and compliance products**, whose purchases are hardest to judge from the outside.
- **Educators and consultants** who want a contemporary case and frameworks to teach with.

No programming knowledge is assumed. Technical details appear only where they change a decision, and are explained when they do.

How to use this book

Each chapter follows the same pattern:

1. **An opening question** that the chapter tries to answer.
2. **Theory:** the research and ideas that bear on the question, cited so that you can read further.
3. **The AttestOS case:** what the case shows, and what it does not.
4. **A framework or tool** you can apply.
5. **Points of debate:** where the argument is contestable.
6. **Discussion questions** for a reading group, a class, a leadership team or your own reflection.
7. **Apply it:** a short exercise for your own products.

The chapters can be read in order or by part. Groups meeting weekly can take one part at a time over five sessions. Chapter 25 collects every framework in one place. Chapter 26 is a facilitator's guide, Chapter 27 is a glossary and Chapter 28 lists the references.

The structure

- **Part I · The new illusion** (Chapters 1–4) explains why products can now look finished without being finished, defines simulated completeness, and shows why it matters most for products whose quality buyers cannot see.
- **Part II · Anatomy of a convincing demo** (Chapters 5–9) dissects the forms the gap takes: fabricated numbers, theatrical buttons, controls that do not control, references to things that do not exist, borrowed authority and tests that test nothing.
- **Part III · The mechanism audit** (Chapters 10–14) sets out six probes that expose those gaps, shows what an outsider can find in an hour, treats claims about third parties, and helps a leader read an audit and decide whether to repair or rebuild.
- **Part IV · Building for proof** (Chapters 15–20) describes how to build so that surfaces stay true: mechanism first, one source of truth for every claim, honest limits, truthful outreach, honest status labels and verifiability as strategy.
- **Part V · Leading AI-built work** (Chapters 21–24) turns to governance: evidence contracts for agents, metrics that resist gaming, law and ethics, and a ninety-day programme.

The case in one page

AttestOS protects website forms (sign-up, login, contact) from automated abuse. A small script on the form has the visitor's browser solve a short computational puzzle. The browser receives a token, which the website's server checks with AttestOS before accepting the form. The check has the same request and response shape as the industry's most common verification service, so websites can switch with little effort. Paid modules add:

- risk signals;
- device tokens that let some phones and computers pass without solving;
- passkeys;
- "AgentPass", a policy for AI crawlers that allows, charges or blocks them.

Prices run from free to $299 a month.

The first version was committed in eleven commits over two hours on 26 October 2026. It had a landing page with live demonstrations, an owner "command centre" reporting nearly 1.5 million requests inspected and $482.16 of revenue, a customer portal, product covers, manuals, downloadable integrations, a vulnerability scanner and an "autonomous scout" that drafted security notices to companies.

The audit, on 27 October, found 65 problems:

- Every number in the console was typed in by hand.
- The security check could be bypassed in seven ways, all confirmed against the live service.
- Nobody could get a key or pay.
- The scanner showed the same "vulnerable" verdict for any address.
- The WordPress plugin locked site owners out of their own sites, and its script came from an unregistered domain that anyone could buy.
- The scout's security notices were built from a scrambled version of each company's domain name, not from any scan.

The rebuild, committed late on 27 October, replaced the whole service with one that works:

- real accounts and keys;
- verified tokens that can be used once;
- a scanner that actually fetches the page;
- outreach that says only what was observed on the recipient's own website, and is held until the legally required postal address is set.

Six follow-up commits on 28 October fixed eleven smaller issues. Six of those were, again, places where words had run ahead of the machinery.

At the end of the window the service was live, with one account (the owner's), ten prospective customers identified and not yet contacted, no customers and no revenue. That is the whole case. The rest of the book asks what it teaches.

PART I

The new illusion

Proof of Mechanism

1 · The demo that wasn't

> **Opening question.** How can a product look finished, convincingly enough to persuade a careful person, when almost nothing behind it works?

On the afternoon of 26 October 2026, a public software repository received its first commit. It was titled "initial production release v2.0.0". Over the next two hours ten more commits followed:

- an owner "command centre";
- a landing page with brand assets and free downloads;
- a paying-customer portal;
- product covers, "live screenshots", user guides and manuals;
- a design system and a footer borrowed from a sister venture;
- an "Autonomous Scout & First-Touch Inbound Mesh";
- the enforcement of a "Customer-in-the-Loop requirement";
- the provisioning of a "VIP Enterprise tenant" with "1-click credentials".

By the end of the afternoon the product, a service for protecting websites from bots, had everything a buyer would expect to see.

This chapter tells the story of what was behind it, and asks how something so empty could look so full.

What a visitor saw

A visitor to the AttestOS website saw a confident, well-designed security product. The landing page offered a five-second scan of any website, live demonstrations of an invisible bot check, and four product modules with their own covers. It also had calculators showing the money a site could save or earn, and downloadable plug-ins for popular platforms.

The owner's console was more impressive still:

- 1,489,204 requests inspected and 142,380 blocked;
- 312,450 hardware attestations;
- 84.6 hours of attackers' computing time "burned";
- $482.16 of revenue from AI crawlers, up 12.4%;
- a status line reading "Whop Webhooks: Connected (2 Plans Active)", meaning that the payment platform was wired in and two plans were selling.

The customer portal told a paying customer they had used 124,180 of their 500,000 monthly verifications and earned "$42.80 AI micropayments". When they saved a policy change, it reported that the change had "Propagated… in <0.03ms".

The repository carried a badge reading "Tests 27/27 passing". The manuals described an auto-tuner, an emergency lockdown and exportable audit logs.

What was behind it

The next day the owner asked for a deep-dive audit. The auditor read every file and confirmed that the live service was byte-for-byte the code in the repository. They then probed every public address and sent test requests to the live service. They also checked whether the domains, software packages and payment links the product referred to actually existed. The audit logged 65 findings. Here are some of them.

Every number in the console was typed in. None came from a database; there was no database. The console had no login. It was pre-filled with a "master key" that also appeared in the page's source code, so every visitor saw the owner's view.

The buttons performed. "Emergency Lockdown" changed a badge on the screen. "Provision tenant" added a row to a table, with a randomly generated key that existed nowhere else. "Dispatch first touch" printed "✓ Dispatched … via Cloudflare Edge". Nothing was stored and nothing was sent.

The security check let almost anything through. Seven separate bypasses were confirmed against the live service. For example, the service accepted any value at all in a field meant for a hardware security token, and it admitted any request that claimed to come from an AI search engine.