THE OPERATOR'S BOOKS

The Point of Failure: An Operating Manual for the Person Running the Estate

Courage Garikai — Coldcode Si Labs, Harare, Zimbabwe ·
Coldcode Si Labs — Harare
First edition, November 2026

Front matter

About this book

This is a manual for one person: the operator of a business small enough that they are the only human in it, and automated enough that the business keeps working while they are not there. It is written for the person who owns the consequences of machine work they did not perform, and who has discovered that the machine has a failover everywhere except the one place that matters.

It is not a book about productivity. Productivity books assume the work is a fixed quantity and the person a variable to be optimised. This book assumes the opposite: the work is unbounded, the person is finite, and the only way to hold both is to design the business so that the finite thing is never asked to be infinite.

The Point of Failure

An operating manual for the person running the estate

by Courage Garikai

Published by Coldcode SI Labs
First edition, 2026

Copyright © 2026 Courage Garikai. All rights reserved.

No part of this publication may be reproduced, distributed, or transmitted in any form or by any means without the prior written permission of the publisher, except for brief quotations in reviews.

Coldcode SI Labs Library · couragegarikai@atomicmail.io

How to read it

If you have an hour, read Part One and Part Three. That is the diagnosis and the bad day.

If you have a day, read it straight through, then fill in the instruments in Appendix B. The instruments are the book; the prose is the argument for them.

If the week is already bad, start at the bad-day protocol in chapter 16, do the three steps, and come back when a day has room in it. There is no penalty for arriving at the book sideways.

If you are reading this for someone else, chapter 23 is addressed to you, and it is four paragraphs.

Each part opens with a statement of what it is for and closes with the thing you should be able to do afterwards. The activities in Part Six are not homework; they are the parts of the method that a reader has to perform once in order to own.

A note on evidence and on what this is not

This book is not therapy and it is not a substitute for care from a professional. It has nothing to say about clinical depression, anxiety disorders, trauma or grief, and no amount of system design will make it say anything. Chapter 22 says where the book stops and a person has to start, and that chapter is not decoration; it is the one that matters if the earlier ones have stopped helping.

What the book does claim is narrower and testable: that a particular way of working — one person, a machine that acts at all hours, an unbounded surface, one decider — produces a predictable kind of wearing-down, and that predictable things can be designed against. The nine rules are the design. They are operational, they are checkable, and none of them requires motivation.

The evidence is one estate: seven ventures, twenty-two works at the time of writing (twenty-four including the two new books in this series), five publishing rails,

an hourly monitor, a three-touch outreach sequence, and roughly three hours twenty minutes a week of human attention, of which about seventeen minutes is genuinely manual. One estate is a case, not a study. The rules are stated in a form you can falsify in your own operation within a month, and Appendix C is the form on which to do it.

Part One — The failure that is not the server

What this part is for: to establish that the operator is a component, with the same obligations as any other component, and that the obligations have been neglected. By the end you should be able to say which single part of your business has no failover, and why.

1. Everything here is redundant except the person

This book was written from inside a working estate: seven ventures, twenty-two works, five posting rails, a queue that refills itself, deploys that verify their own work, checkouts that take money at three in the morning without asking anyone. Every part of that was built for one reason, and it was not efficiency. It was so that one person could run it without being destroyed by it.

Inside that system, almost everything has a failover. The press rail retries. The bank refills before it empties. A deploy that lands wrong is redeployed and verified. A worker that returns a 404 at the front door is caught by an hourly check that says so in plain language. If a credential expires, the run says *credential missing* instead of going quiet. The design assumption is that any single component can be absent for a while and the estate keeps standing.

There is exactly one component with no redundancy at all. It is not the server. It is the operator.

> Figure 1 sets the two columns side by side: six components that fail and recover, and one that fails and does not.