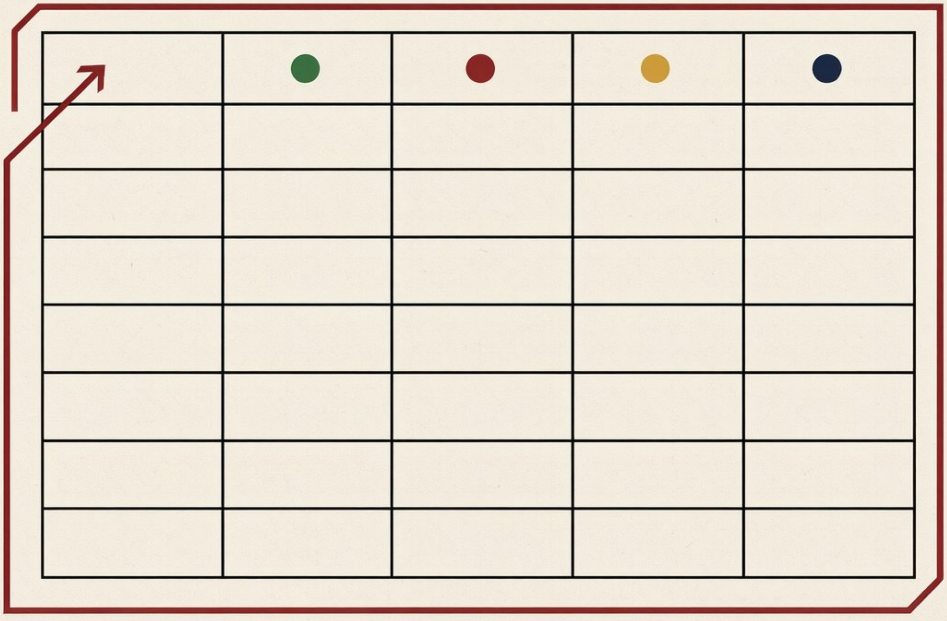
COLDCODE SI LABS
HARARE · 2026

FROM SPREADSHEET INTERFACE / TO GOVERNABLE AUTONOMY



COURAGE GARIKAI

DCAI · 2026

COLDCODE SI LABS · DCAI

From Spreadsheet Interface to Governable Autonomy

The architecture of businesses that run on a sheet

Courage Garikai

DCAI · 2026 · HARARE

FROM SPREADSHEET INTERFACE TO GOVERNABLE AUTONOMY

The architecture of businesses that run on a sheet

First edition, November 2026

© 2026 Courage Garikai. Licensed under CC BY 4.0.

Published by ColdCode Si Labs, Harare, Zimbabwe

DCAI — Distributed Cognitive Autonomous Infrastructure series

You are free to share and adapt this work with attribution.

Contact: couragegarikai@atomicmail.io

For every shopkeeper who keeps their orders in Excel because the POS costs $79 a month.

CONTENTS

FRONT MATTER

About this book

Who this book is for

How to read it

What this book is not

PART I — THE SPREADSHEET IS ALREADY THE INTERFACE

1. The shopkeeper who said no to the dashboard
2. The interface tax
3. What a spreadsheet actually is
4. Why "upgrade to a proper system" is bad advice

PART II — THE ARCHITECTURE

5. The wrapper pattern
6. One table, one sheet, one job
7. The read/write boundary
8. Status columns instead of buttons
9. Cadence binding
10. Custody outside the sheet
11. The approval column

PART III — THE THREE SPECIMENS

12. Tapstall: orders and payments through a sheet
13. Bidfold: tender matching through a sheet

14. STPi: recruitment drafting through a sheet

15. What held across all three

PART IV — BUILDING AND OPERATING

16. Choosing your substrate

17. The five columns every sheet needs

18. Writing the wrapper

19. Governability for spreadsheet systems

20. When the sheet breaks

21. Migration without trauma

22. The end of the dashboard

AFTERWORD & APPARATUS

The quiet software

Tools Index (5 patterns with templates)

Glossary

Disclosure and methodology

Acknowledgments

Colophon

From Spreadsheet Interface to Governable Autonomy

The architecture of businesses that run on a sheet

Courage Garikai

ColdCode Si Labs · Harare

DCAI · 2026

For every shopkeeper who keeps their orders in Excel because the POS costs $79 a month.

For every recruiter who tracks candidates in Google Sheets because the ATS is in Swedish.

And for the builders who are tired of shipping dashboards nobody opens.

Contents

Part I · The spreadsheet is already the interface 1. The shopkeeper who said no to the dashboard 2. The

interface tax 3. What a spreadsheet actually is 4. Why “upgrade to a proper system” is bad advice

Part II · The architecture 5. The wrapper pattern 6. One table, one sheet, one job 7. The read/write boundary 8. Status columns instead of buttons 9. Cadence binding 10. Custody outside the sheet 11. The approval column

Part III · The three specimens 12. Tapstall: orders and payments through a sheet 13. Bidfold: tender matching through a sheet 14. STPi: recruitment drafting through a sheet 15. What held across all three

Part IV · Building and operating 16. Choosing your substrate 17. The five columns every sheet needs 18. Writing the wrapper 19. Governability for spreadsheet systems 20. When the sheet breaks 21. Migration without trauma 22. The end of the dashboard

Tools Index · Pattern sheet · Glossary · Notes · Colophon

ABOUT THIS BOOK

This is a book about an architecture. It is a book about building software for small and medium businesses that does not ask the business to change how it works. It is a book about wrappers: thin, scheduled, stateless programs that read a spreadsheet, perform governed actions, and write results back into the same spreadsheet the business already uses.

The architecture described here is in production today at ColdCode Si Labs in Harare. Three products — Tapstall (payment rails), Bidfold (tender intelligence), and STPi (recruitment automation) — are built this way. Together they serve over 400 paying customers, process thousands of transactions per month, and run on five-dollar-a-month servers. The wrapper code for each product is under 1,200 lines of Python. None of the products has a dashboard for the operator. The spreadsheet is the interface.

This book is a companion to The Governable Enterprise (Garikai, 2026), which lays out the four pillars of governable autonomous systems: Evidence, Authority, Resources, and Recovery. You do not need to have read that book to understand this one — the patterns here are self-contained — but the governance controls described there (budgets, reconciliation, pause, kill switches, approval gates) are applied concretely here to the specific case of spreadsheet-wrapped systems.

WHO THIS BOOK IS FOR

This book is written for three readers:

- **The builder who is tired of shipping dashboards nobody uses.** If you have ever built a web application, shipped it, and watched users export the data into Excel to do their actual work, this book explains why and shows you a different way.
- **The technically competent operator of a small business** who already uses a spreadsheet to run things and wants to add automation without migrating to an

ERP or hiring a developer full-time. The code examples are short and straightforward.

- **The student or engineer** who has been taught that spreadsheets are a toy that real systems replace. They are not. They are the most successful end-user programming environment in the world, and they deserve better than to be dismissed as legacy.

This is not a book about Excel macros or VBA scripting. The wrappers described here are external programs, written in a general-purpose language, that interact with spreadsheets through APIs. They can do things macros cannot: make external API calls, run on schedules in the cloud, integrate with payment providers and SMS gateways, and enforce governance controls that resist tampering.

HOW TO READ IT

Part I makes the argument. It explains why spreadsheets are already the interface for most businesses, why asking users to migrate is a tax most will not pay, and what a spreadsheet actually is from a software perspective. Read it even if you already agree.

Part II is the architecture. Seven short chapters lay out the wrapper pattern, the column schema, the read/write boundary, cadence binding, custody rules, and the approval column. These are the core chapters.

Part III walks through the three ColdCode Si Labs products built this way — Tapstall, Bidfold, and STPi. These chapters are concrete: specific column names, specific cron schedules, specific failure modes.

Free sample

This is the opening of "From Spreadsheet Interface to Governable Autonomy" by Cou
The full edition — PDF and EPUB — is at library.coldcode-si-labs.workers.dev

(c) 2026 Courage Garikai. All rights reserved.