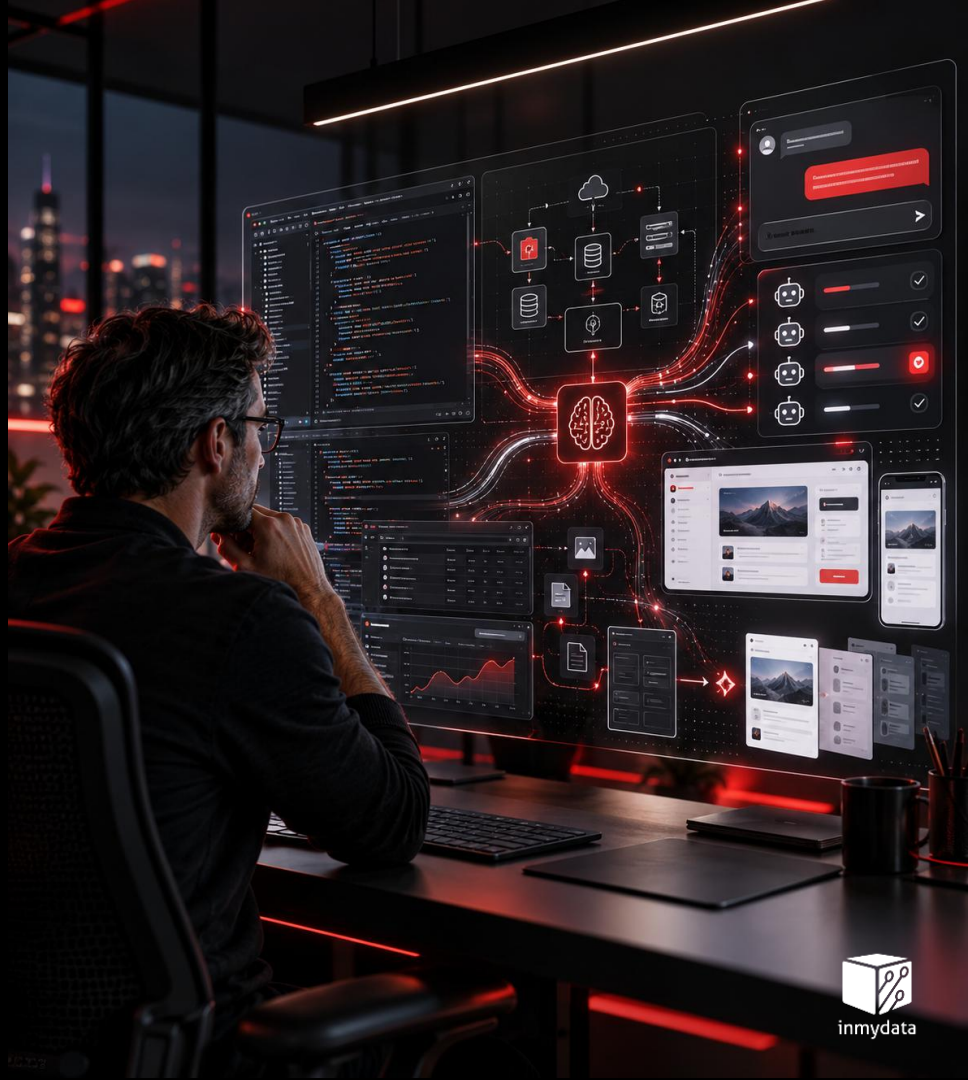
AI-First
in Practice.

Nick Finch
Chief Technology Officer, inmydata

PUG Challenge 2026



“The greatest danger in times of turbulence is not the turbulence. It is to act with yesterday’s logic.”

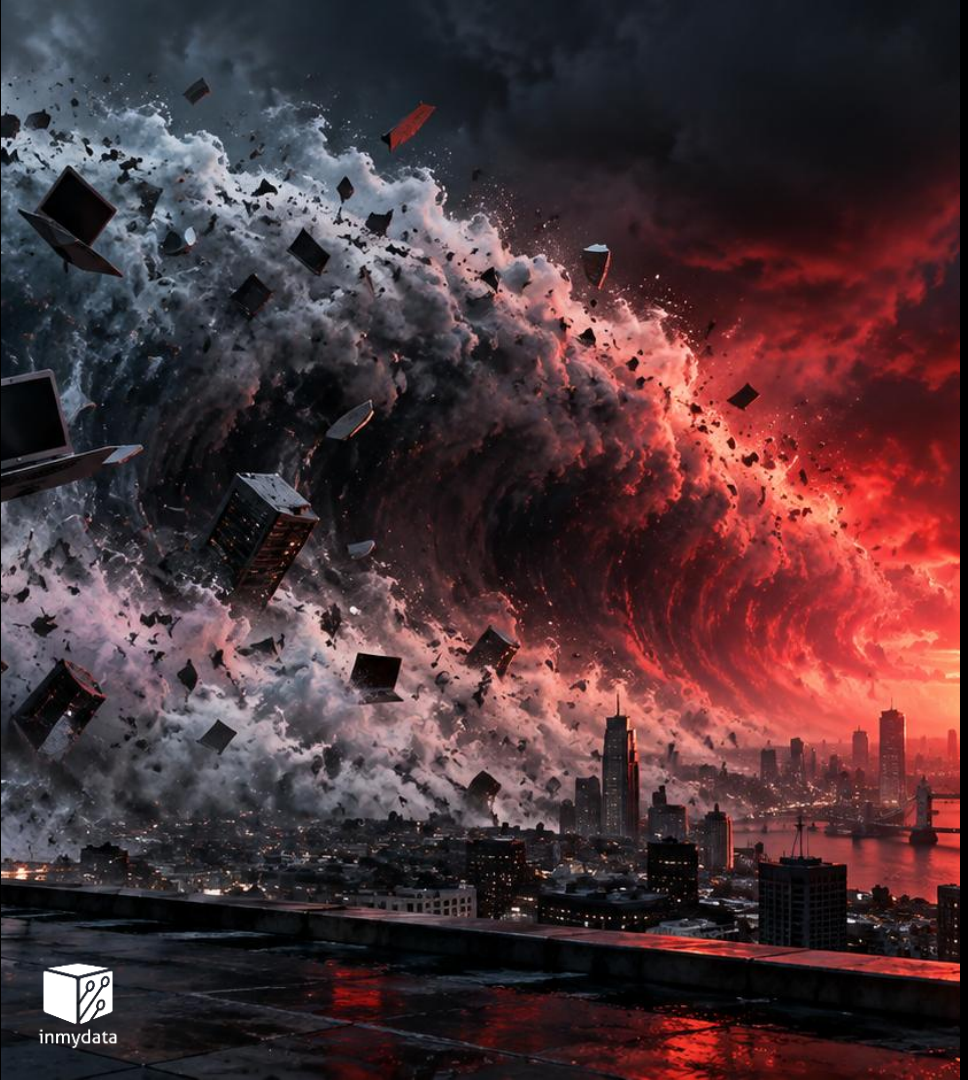
Peter Drucker
management consultant



Yesterday's logic was a human at a screen,
and one product, off the shelf, for everyone.

This is a talk about what replaces both. Where to look, three systems we've built on data you work with every day, and a way of building that changes what's worth building.

inmydata



$2 trillion

The SaaSpocalypse

February 2026. Around $285bn wiped from SaaS valuations in 48 hours.

Within weeks, roughly $2tn gone from software stocks. Sector down 20%+ while the S&P 500 sat near an all-time high.

The trigger wasn't a model. It was an agent product that could do the knowledge work.

Bloomberg, Feb 2026. J.P. Morgan via Fortune, Feb 2026. Reuters, Feb 2026.

Seats

A seat is a licence for a human to sit in front of a screen

Atlassian fell two-thirds on fears of seat compression, then cut 10% of its staff to fund AI.

Workday has cut its workforce twice this year, most recently at the end of September.

The market is pricing in fewer seats. Fewer seats means fewer screens.

Atlassian and Workday SEC filings, 2026.



inmydata

Until four years ago, we were an analytics vendor

Twenty years. Three hundred plus customers, a lot of them in this room's world.

Our moat was the dashboard builder. 39 dashboard objects, each with its own screen and hundreds of options. A day's training just to know what was possible.

It was what we demoed, what we sold on, and what we believed nobody could catch up with.





A year ago we replaced it

inmydata Studio. An agentic dashboard designer.

The user talks their way to a dashboard. No training. No maze.

Built in two weeks.

inmydata

If we could build it in two weeks, so could anyone.

The UI we had spent years trading on was suddenly worth close to nothing. What was underneath it was not.



What the agent actually needed

Metadata

Our business logic. The layer that makes structured data in an ERP or operational application understandable to a model.

Speed

Conversations don't survive a thirty-second query. Millions of rows in a fraction of a second.

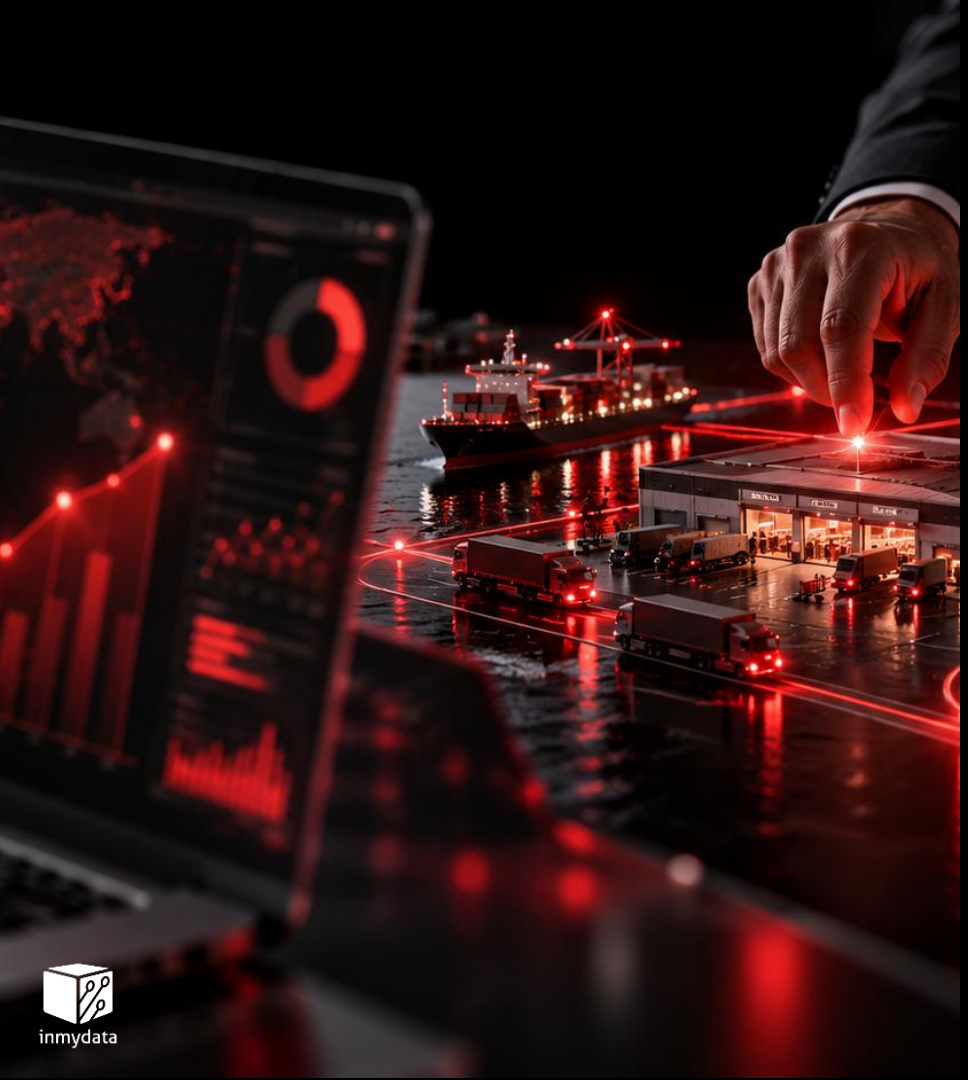
Security

A mature, sophisticated data governance layer. An agent that can see everything is a liability, not a product.

inmydata

AI changed the UI.
It didn't change the logic, the architecture or the governance.
That's the moat.

inmydata



Why do people look at dashboards?

We had the holy grail. Real self-service analytics.

But a dashboard is a proxy. People want a picture of the business so they can act.

The dashboard was never the point. The action was.

inmydata

If a system has the figures and the judgement,
nobody needs the dashboard.

So, we judged the days of business analytics were numbered and started building the systems that act.

inmydata

840 manager days a year,
spent preparing for a conversation.

Fraser Hart and Fields. 35 jewellery stores.

Every period, every store manager reviews every colleague, and each spends two days pulling the figures from our dashboards first.



The agent prepares the review

The data

Five years of their sales and KPI history, already in our platform. No integration project.

The judgement

Their People Director's review process, captured through an expert interview. What good looks like, what to raise, how to say it.

The outcome

The agent analyses and drafts. The manager reviews, edits and owns the conversation. No dashboard in the loop.

inmydata

Buying green coffee

Segafredo Zanetti Nederland. A roaster buying a volatile commodity months ahead.

Supply depends weather, crop yields, transport costs. Demand sits in their own ERP.

The people who do it well hold all of that in their heads.



An expert buying system

Inside

Stock, contracts and sales history from the Agresso ERP.

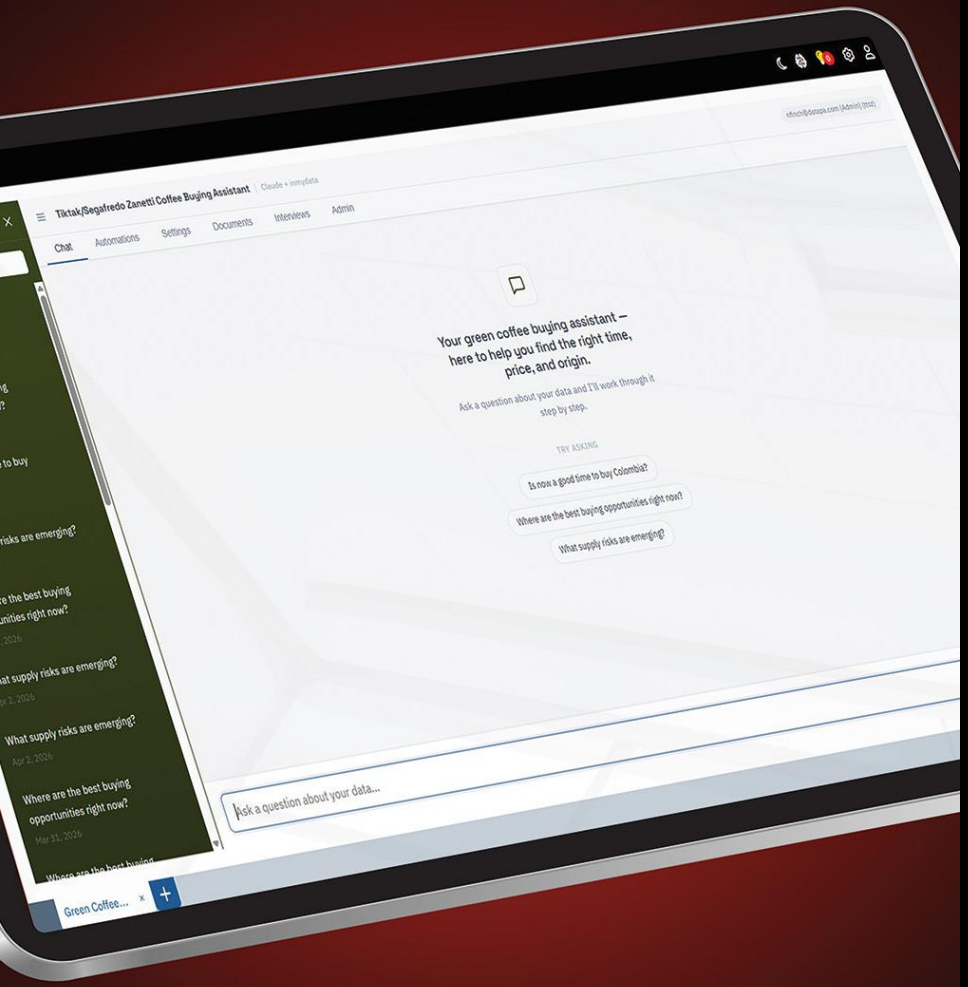
Markets

Live arabica and robusta futures from Barchart.

The world

NASA POWER weather across 23 producing countries. ENSO forecasts. Conab harvest data from Brazil.

inmydata



What changed

The buyers ask. “What's our exposure if arabica moves ten percent?”

They get an answer they can interrogate.

The domain logic didn't change. Their expertise didn't change. We packaged it for an agent.

inmydata

A window on the database

White Star Software. The people you call when a Progress OpenEdge database is on fire.

Protop a monitoring product with dashboards, alerts, diagnostics. A window on the database for an experienced DBA.

The value was never the window. It was the DBA looking through it.





Giving the agent what the DBA know

Build an expert system by interviewing the DBA experts and ingesting documents.

Combined Protop's own monitoring tools so the agent sees the same numbers the DBA sees.

Tacit expertise becomes something an agent can act on.

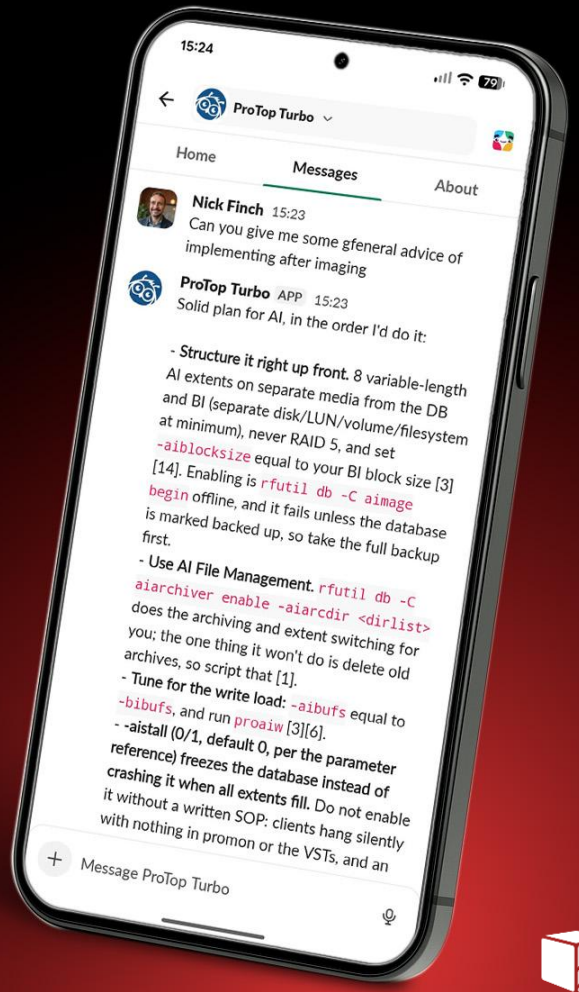
inmydata

From a window to a caretaker

Stage one. The agent offers expert advice inside the product.

Stage two. A trusted path for the agent to act, and fix the issue itself.

The value is the knowledge and the tools, combined with an agent. Not the UI.



inmydata

The pattern to all three

Take what's valuable

Domain logic. Proprietary data. Governance. The encoded expertise of everyone who ever worked on it.

Expose it to agents

Exposed the logic through an API or MCP server an agent can call. Made it fast. Made it governed. Built the system that acts, not the screen that shows.

Replaced what became irrelevant

Most of the screens. Most of the training. Any moat built on how hard a product is to replicate visually.

inmydata

All were built at a pace that would have been impossible twelve months ago.

Being AI-first doesn't just mean building AI solutions. It means building with AI. And that changes what's worth building.

inmydata

The tools aren’t the variable.
The discipline is.

45% of AI-generated code samples contain security flaws. AI co-authored pull requests carry 2.7 times the vulnerabilities. The horror stories are real.

They are stories about undisciplined use. Prompt, accept, ship. No architecture, no review, no understanding of what was built.

DORA 2025: AI is an amplifier. Disciplined teams pull ahead. Fragmented teams accelerate into instability.

Veracode, 2025. CodeRabbit, 2025. DORA, 2025.



inmydata

THE WORKFLOW

This is what discipline looks like

SPECIFY

1 — Interview the design
One product question at a time, checked against the record and the code.

2 — Adversarial tear-down
Clean-context model attacks the spec: gaps, security holes, blind spots.
↻ until objections are housekeeping

3 — Phased implementation plan
Clear scope per phase, with test definitions written into each phase.

4 — Tear down the plan
Same bar: attack and refine before a line of code exists.
↻ until objections are housekeeping

BUILD · REPEATED FOR EVERY PHASE

5 — Build the phase
Coding agents implement against the plan; parallel across modules.

6 — Test until green
Run the phase's test definitions; fix and rerun until all pass.
↻ until all tests pass

7 — Commit the phase
Triggers automated code review + Claude Code Security check.

8 — Fix & commit findings
Address every recommendation, then loop to the next phase.
↻ next phase → step 5

RELEASE

9 — Deploy to staging
Automated pipeline; push to staging in under 40 minutes.

10 — Penetration test
OWASP ZAP + Nuclei suite against the deployed instance.

In practice

Commit cadence flexes with phase complexity, and staging deploys can happen mid-build where features need a live instance to test.

inmydata

THE DEVELOPMENT ENVIRONMENT

Encoded, not remembered

Every Claude Code session at inmydata loads a curated skills library, part built in-house, part adopted from third-party best practice, that governs how agents plan, build, review, test and ship.

Planning & Specification
6 skills

Implementation & Workflow
9 skills

Code Review & Quality
10 skills

Testing & Verification
4 skills

Frontend & UI
6 skills

Backend, APIs & Python
10 skills

Observability & Ops
6 skills

Docs & Project Setup
5 skills

Codebase Understanding
8 skills

60+ skills · shipped as Claude plugins from GitHub



inmydata-skills:document

Every module and function carries a defined header: its dependencies, and what depends on it. The agent knows the blast radius of a change before touching it. Fewer bugs, lower token usage, a faster agent.



What one looks like

name: document
description: Add or update structured documentation headers on source files in the inmydata stack for AI agent navigation. Invoke on a file, directory or glob.
Frontmatter says when to use it; the body says how. Matched against the task in every session.



What that looks like for a small team

14

production systems shipped in the last year. The first five alone were roughly two years of traditional development.

50x

code output this year, measured off the repositories. Twelve times the commits, absorbed without chaos because the gates were already there.

< 40 minutes

from push to staging through a fully gated pipeline. Automated review, security scan, test suites. Several times a day.

inmydata

WORKING AT AGENTIC VELOCITY

Modular by design. One developer per module.

THE NEW BOTTLENECK

At agentic velocity, merging becomes the most expensive step.

When one developer can generate in days what used to take a team months, concurrent changes to a shared codebase turn every merge into a costly reconciliation exercise.

OUR ANSWER

Containerized development

Every project is modular by design, with clean interface contracts between modules.

One developer per module

Only one developer works inside a module at a time. No merge tax, ever.

Parallelism across modules, not within them

A developer plus agents produces what a team once did, so the constraint costs nothing.

Modules over shared services is the shape agentic development wants.

A product built as independent modules over shared services is not just architecture. It is the unit of assignment: one developer plus agents inside each module, and the interface contract is the only thing two developers ever have to agree on.

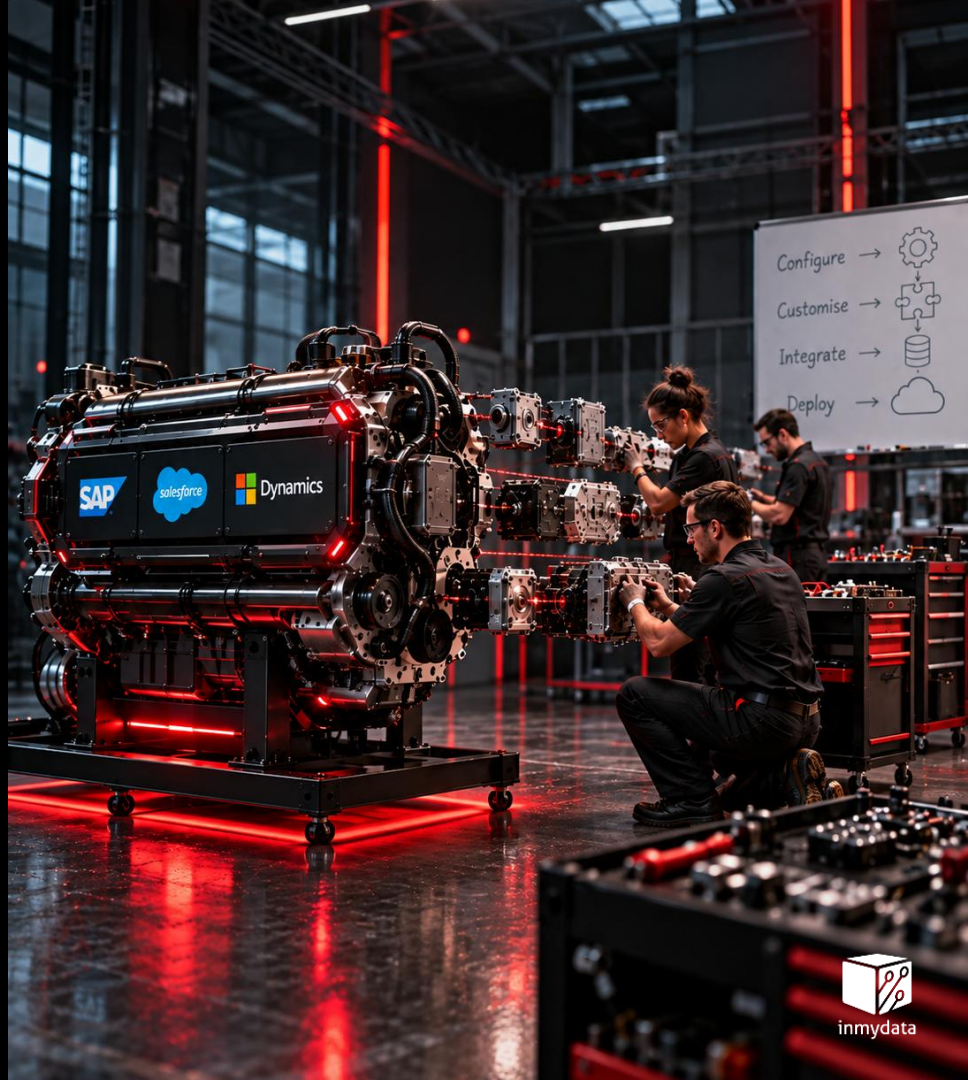
inmydata

The product was always a ratio

SAP, Salesforce, Dynamics didn’t sell finished software. They sold a configurable core, and an industry of consultants did the fitting. On most rollouts the fitting was the bigger invoice.

The product is the part you sell repeatedly without rebuilding it. The service is everything you redo per customer.

Every software business, including yours, is that ratio.



The line has moved.

The code is now the per-customer part.

The process is the product.

Every line of code we write for a customer belongs to the customer. What we keep is the framework it was built with. That is what we sell twice.



Don’t build a single product for every customer

Build the framework that fits your product to every customer. Thirty years of core.

A process that generates the fitting at agent speed. Bespoke, for every customer, at a price they’ll pay.



AI-first for OpenEdge developers

Your core

Business logic. Data. Governance.

Thirty years of it, already built, already trusted.

Expose it

APIs and MCP servers.

The authoritative source every agent calls into.

Agentic delivery

Specify, build, deploy, at pace. A process to build agents.

The fitting, done at agent speed.

Apps that act

Bespoke agentic solutions for each customer.

In weeks, at a price they will pay. Systems that act, not screens that show.

What you keep, and sell to every customer

What each customer gets

inmydata

Yesterday's logic was the screen, and the product on the shelf.
The data you already have is enough to begin.

Find one opportunity. Scope it tightly. Start today.

inmydata

Thank you

Nick Finch
nfinch@inmydata.com
inmydata.ai

