

Familiar

Agent Native Compute

Turning senior engineering capability into scalable,
agentic delivery

The problem

Small businesses don't have a lack of software.

They have a lack of **capability**.

- Systems are fragmented
- Important knowledge lives in people's heads
- Integration work is expensive
- Automation is brittle
- Senior technical expertise is scarce
- AI can generate code, but doesn't solve the operating problem

The constraint isn't software.

It's the ability to make software work.

The insight

Agents don't primarily need to become smarter.

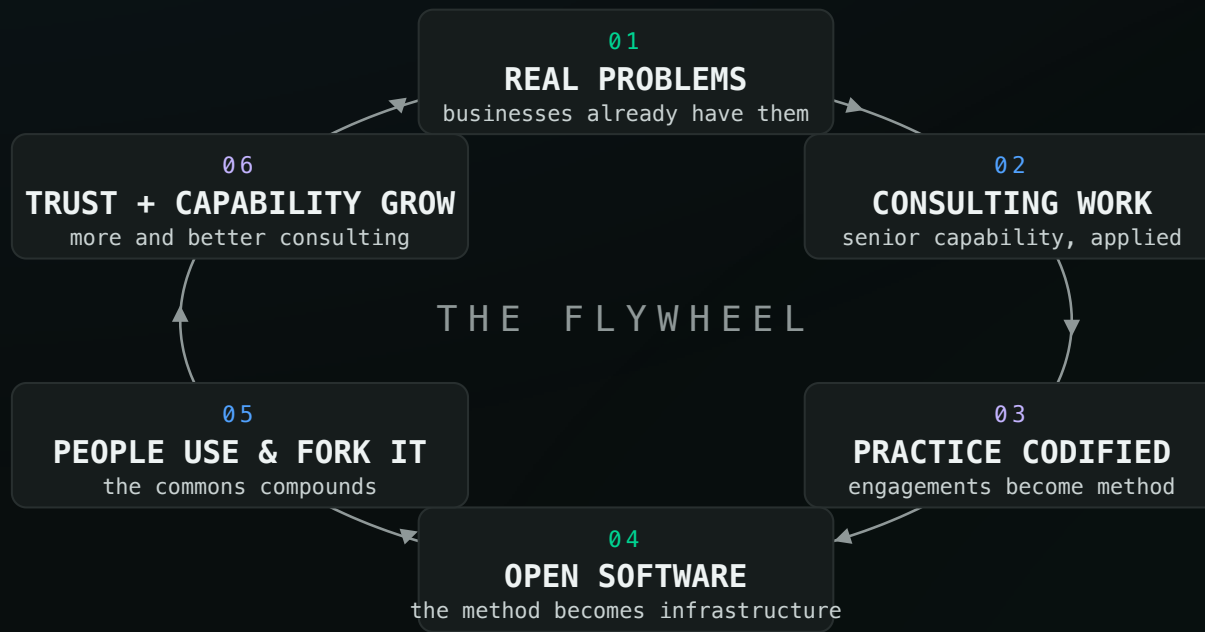
They need an environment in which their intelligence can **compound**.

- persistent context
- coordination
- bounded authority
- accessible knowledge
- observable work
- executable proof
- reliable handover

Familiar is an attempt to build that environment.

I. The conceptual flywheel

The loop is the business — every slide in this deck is a snapshot of it turning



The estate is already here

Fifty-plus repositories across seven capability regions — the flywheel is already turning

COORDINATION

Chorify

control plane →
execution swarm → blackboard

MEMORY & CONTINUITY

Relay

state handover

Loom

how agents think

WHERE AGENTS RUN

Hearth

where
agents run

BEHAVIOUR & PROOF

Behave

executable scenarios

CONSTRAINTS & AUTHORITY

Boundary

permissions

Verdict

adjudication

SHARED WORLD

Fabric

shared state

DATA

Ground

DuckDB-backed · JSON / YAML /
TOML / CSV / Parquet

Den

agent
storage

Gym

simulation

Markdown

documents as
first-class data

ERP STACK

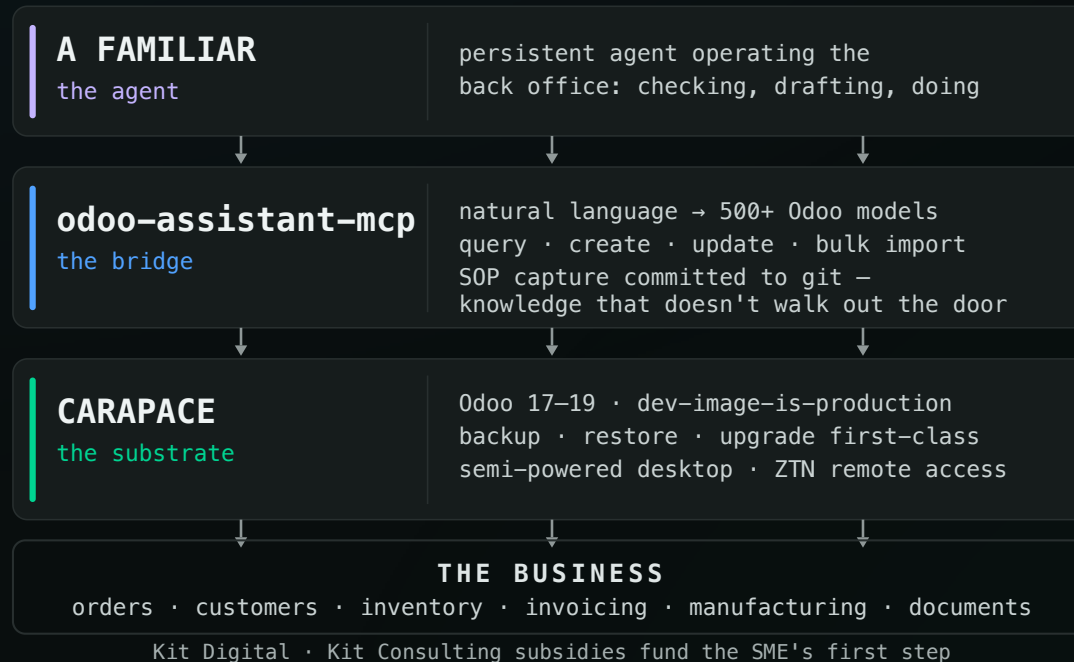
Carapace

the Odoo
substrate

50+ repositories · seven capability regions · MIT on everything released

The agent-driven ERP stack

Old proof, new discipline — the MCP shipped before the practice was codified



The first commercial wedge

Make existing businesses agent-native.

Start where the economic value is obvious:

SME operations

ERP · finance · inventory · purchasing · production · documentation ·
customer operations

Familiar provides

Senior engineering capability + agent labour + proven open infrastructure

rather than asking a 20-person company to hire an AI team.

Carapace

Agent-driven ERP for small businesses

Odoo provides the business system. Familiar provides the agentic operating layer around it.

Carapace

- production-ready Odoo environments
- reproducible infrastructure
- deployment & operations
- upgrades, backup & recovery
- open-source foundation

Odoo Assistant

Agents can:

- query business data
- create and update records
- perform bulk operations
- follow documented procedures
- capture operational knowledge into the repository

Don't replace the ERP. Give the business an operator.

Why Spain?

The demand is real, measured, and already subsidised.

99.8% SMES

I

77% of Spanish firms have fewer than five employees — Bank of Spain.

-7% PRODUCTIVITY

II

Spain runs ~7% below the EU average — OECD 2024.

8.7% AI ADOPTION

III

Against 49.2% of large enterprises — DESI 2024.

€24K + €3-29K

IV

Kit Consulting and Kit Digital pay the SME's first step — Red.es.

12.6% OF GDP

V

Tourism — the wedge's first market — INE.

Spain: the wedge inside the wedge

Hospitality

Tourism is one of Spain's defining economic sectors — and hospitality contains exactly the kind of operational complexity Familiar is designed for.

Restaurants

POS · inventory · suppliers · purchasing · staffing · rotas · accounting

Hotels & guesthouses

Bookings · procurement · housekeeping · maintenance · accounting
· operations

The opportunity

Don't sell "AI". Sell:

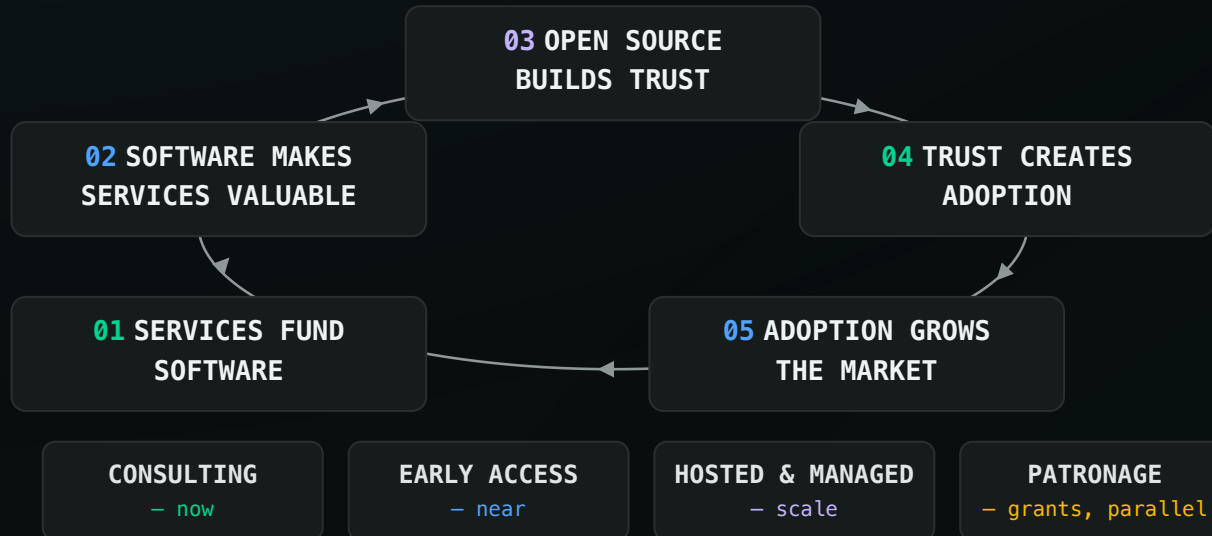
Less administration. Fewer disconnected systems. Better operations.

And make the agent the mechanism rather than the product category.

II. The economic flywheel

One loop, seen through the ledger

FOUR ROUTES TO REVENUE



The commercial model

Four routes to revenue.

● NOW

01 — Consulting

High-value transformation and engineering engagements.

● NEAR TERM

02 — Early access

Pre-release Familiar capabilities for selected customers.

● SCALE

03 — Hosted / managed

Familiar-operated infrastructure and agentic business systems.

● PARALLEL

04 — Patronage & grants

Funding the open commons and social programme.

III. The social flywheel

Why opening it doesn't destroy the business

The commercial engine funds a commons. Familiar's open-source strategy is not merely a distribution tactic — it is part of the mission.

• LEGIBILITY

Make advanced technology understandable and inspectable.

• OPERABILITY

Make it possible for ordinary organisations to actually run.

• INTERCHANGEABILITY

Prevent customers becoming permanently dependent on one provider.

• INSTITUTIONALITY

Build something that can outlive its founder.

Take the map. Leave the institution.

Two ways to fund the mission

The question is which combination produces the strongest flywheel — not simply who pays the bill

Option A — Founder funded

JLDB funds the operating company and development programme.

- maximum control
- fastest execution
- simpler governance
- larger initial capital requirement
- founder carries the financial risk

Option B — Co-funded

JLDB + a strategic co-funder

- capital
- customers
- local market access
- sector expertise
- institutional credibility
- training / employment capacity

Capital goes toward — operating team · agent infrastructure · productisation · market entry · training · OSS development.

The company structure

JLDB Ltd

Existing UK company

- founding / funding vehicle
- group assets
- IP and category stewardship
- international consulting

Spanish S.L.

EU operating company

- Spanish tax domicile
- SME engagements
- local employment
- customer operations
- Spanish / EU programmes
- hospitality wedge

The structure keeps the commercial operation close to the market — while preserving the wider Familiar commons.

What needs to be proven

The next phase is **economic, not architectural**. The numbers are the research programme — deliberately not claims in this deck.

Market

- Who buys first?
- How large is the reachable market?
- Which channels actually work?

Economics

- What does a customer cost to acquire?
- What does an engagement produce?
- What is the gross margin of agent-assisted delivery?

Productisation

- Which consulting patterns become repeatable products?

Scale

- How does one operator become a team?
- How much additional revenue does each trained operator unlock?

Funding

- What level of initial capital gets the flywheel turning?

The thesis

Familiar is building a new kind of technology company.

Not an AI agency. Not an ERP vendor.

Not an OSS foundation. Not a consultancy with a chatbot bolted onto it.

A company where **senior human capability** is amplified by **persistent, agentic software systems** — and continuously codified into **an open, reusable technological commons**.

Familiar

Agent Native Compute

Start with the problems businesses already have.

Use agents to make expert capability scalable.

Turn every engagement into infrastructure for the next.

Keep the resulting knowledge open.