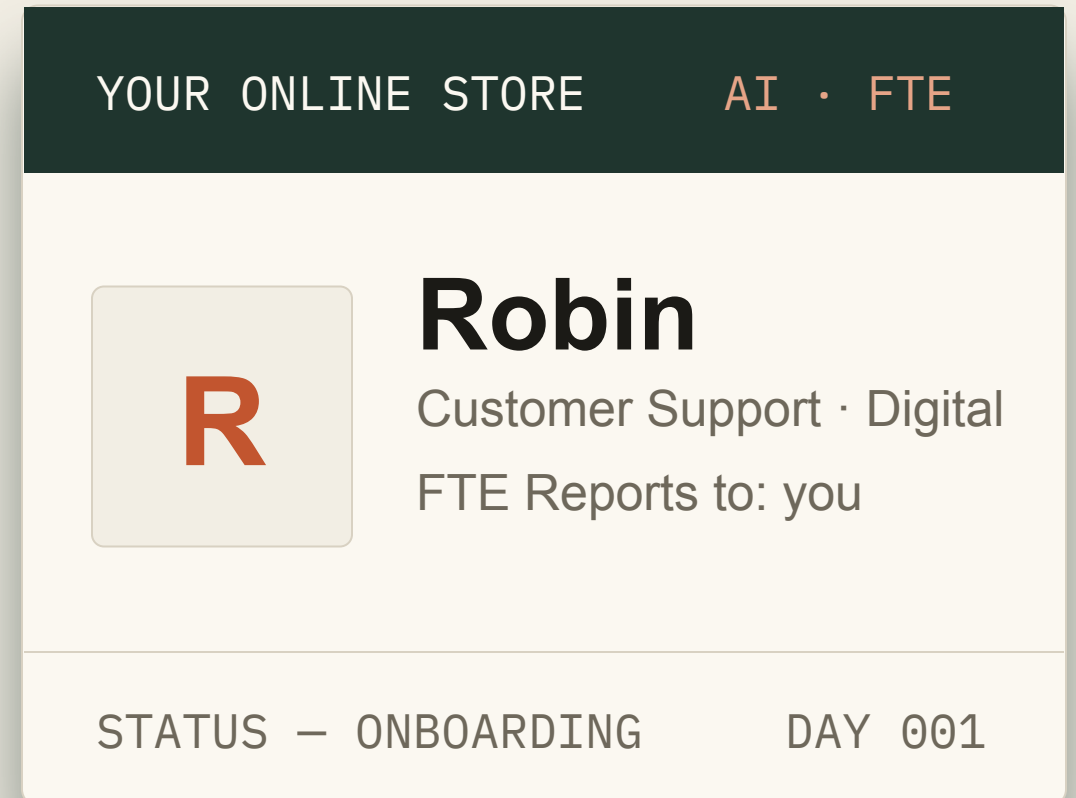


# Hire your first **AI** **employee**

From idea to your first working build — in plain language, no engineering background required.

Building a Digital FTE



YOUR ONLINE STORE    AI · FTE

**R**

**Robin**  
Customer Support · Digital  
FTE Reports to: you

STATUS — ONBOARDING    DAY 001

For everyone — 12th-pass to knowledge worker

## THE BIG IDEA

# Software you **use** is becoming workers you **direct**

---

### FOR DECADES

You open the app. You click the buttons. You do the work; the software just holds it.

---

### NOW

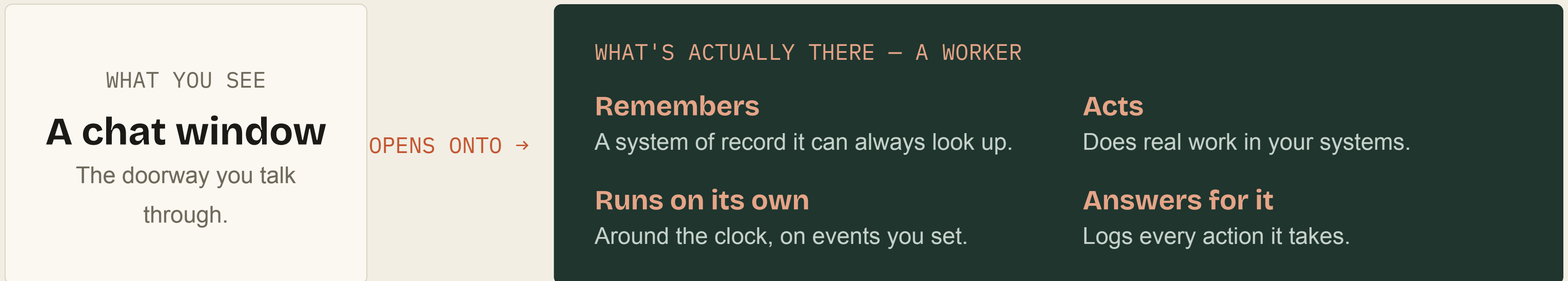
You describe the outcome. A capable worker does the steps, remembers, and reports back.

This guide is about that second kind of worker — what it is, what it's made of, and how you put one to work in a real business.

THE BIG IDEA

# What an AI employee actually is

You reach it through chat, so it's tempting to call it a chatbot. It isn't. Chat is just the doorway — behind it stands a worker that remembers, acts in your real systems, and keeps working when you're not watching. People call it an **AI worker**, an **AI employee**, or a **Digital FTE**.



Named for **how you build it**, **who it joins**, and **what it costs** — an employee you assemble and put on a team, not an app you open.

THE BIG IDEA

# A chatbot answers. An employee gets the job done.

A CHATBOT

AN AI EMPLOYEE

MEMORY

Forgets when the chat closes

Runs on a system of record it can always look up

ACTION

Tells you what to do

Does it in your real systems — looks up the order, issues the refund

INITIATIVE

Only moves when you message it

Wakes itself on events and runs around the clock

ACCOUNTABILITY

No trail of what happened


Logs every action so the company can replay it

Same friendly conversation on the surface. Underneath, three things were added:

**abilities**, **memory**, and a **wire** to your real systems.

THE EXAMPLE WE'LL FOLLOW

YOUR ONLINE STORE    AI · FTE



**Robin**  
Customer Support · Digital  
FTE Hired: today

SHIFT — ALWAYS ON    DAY 001

## Meet Robin, your store's new support hire

**Answers** customer questions in plain language, any hour.

**Looks up** a customer's orders and past conversations.

**Issues refunds** — but only within your policy, and with sign-off.

**Logs every action** so you can always see what happened.

Every idea in this guide, we'll watch Robin actually do.

## THE BIG IDEA

# You don't write the code — you direct it

You won't be typing software. You work through a **general agent** — a capable assistant that builds while you steer. You're the manager; it's the builder.

01

### **Brief**

You say, in words, what you want built next.

02

### **Plan**

It proposes a plan for you to read.

03

### **Approve**

You give it the go-ahead, or adjust.

04

### **Build**

It writes and wires everything up.

05

### **Run**

You watch it work, and refine.

Your job is judgement — what to build, and whether the result is right. That stays human.

# The anatomy of an AI employee

Strip away the surface and every AI employee is the same three parts.  
Once you can name them, you can build one.

## A — ABILITIES

### Skills

The things it knows how to do, picked up only when needed.

## B — MEMORY

### A source of truth

The one place the facts live, that it can always look up.

## C — THE WIRE

### A standard plug

How it safely reaches your real systems and tools.

# Abilities: skills it picks up when needed

An ability isn't baked into the employee. It's a skill it can pick up — a self-contained how-to for one job, that it reaches for only when a task calls for it.

## **ROBIN**

Robin has a skill called "**issue a refund.**" It sits on the shelf until a customer actually asks for their money back — then Robin opens it, follows the steps, and puts it away. Add a "track a package" skill tomorrow and nothing else has to change.

# A skill is just a folder of know-how

## **issue-a-refund/**

- | SKILL.md
- | scripts/
- | references/

### SKILL.MD – THE INSTRUCTIONS

Plain steps, in plain English: how to do the job, when to use it.

### SCRIPTS/ – OPTIONAL HELPERS

Small bits of automation it can run, if the job needs them.

### REFERENCES/ – OPTIONAL DEEP DETAIL

The full policy or examples, opened only when truly needed.

---

Because it's just a folder, a skill can be reviewed, versioned, and shared like any document.

# It can carry hundreds of skills, cheaply

A worker who read every manual before every task would never start. So it reads only the titles on the shelf — and opens a manual when, and only when, the work calls for it.

---

STAGE 1 · DISCOVER

## Read the titles

It always knows the name and one-line purpose of every skill. Cheap, all the time.

---

STAGE 2 · ACTIVATE

## Open the manual

When a task matches, it reads that one skill's full instructions. Only then.

---

STAGE 3 · EXECUTE

## Reach for the detail

It pulls in deep references or scripts only at the moment it needs them.

This is why a single employee can hold a whole library of abilities without slowing down.

# The hard part is teaching it

## when to use one

A skill only fires if its one-line description matches how people actually ask .  
Writing that line well is the real craft — and the part you own.

TOO NARROW – MISSES PEOPLE

**"Use when the user asks to issue a refund."**

---

Never fires on: *"this arrived broken," "I want my money back,"*  
*"cancel order #4471."*

WRITTEN FOR REAL LIFE

**Names the real situations and the words people use — including the ones that never say "refund."**

---

**Robin** now catches the broken-item complaint and the angry "money back" alike.

A good test: delete the obvious word from the description. Does it still say when to fire? If not, sharpen it.

# Memory: a single source of truth

The thing a chatbot lacks. Give the employee one authoritative place the facts live — the way a business already trusts its CRM or its ledger — and it stops guessing and starts knowing.

## **ROBIN**

A customer writes in. Before answering, Robin looks them up: **their last five conversations, their orders, what you already promised them.** Same customer, two weeks later — Robin still knows. Nothing was forgotten when the chat closed, because the truth never lived in the chat.

# The four things an employee remembers

## 01 — Business records

### The operational truth

Customers, orders, tickets — the things you look up and update.

---

**Robin:** "Order #4471 shipped Tuesday."

## 02 — Reference library

### Knowledge it can search

Your policies, help articles, past resolved cases.

---

**Robin:** "Our refund window is 30 days."

## 03 — Live state

### What's happening right now

Which conversations are open, what's waiting on approval.

---

**Robin:** "3 chats open, 1 awaiting sign-off."

## 04 — The trace

### A record of what it did

Every action, in order, so the company can replay it.

---

**Robin:** "Refund of ₹1,200 issued, 4:02pm."

# It searches by meaning, not exact words

Ask *"have we seen this before?"* and an exact-word search fails the moment the words differ. The employee instead finds things that mean the same — even worded completely differently.

## MATCHING WORDS

Customer: "the lid won't click shut"

Past case: "container doesn't seal" → **missed**. No shared words.

## MATCHING MEANING

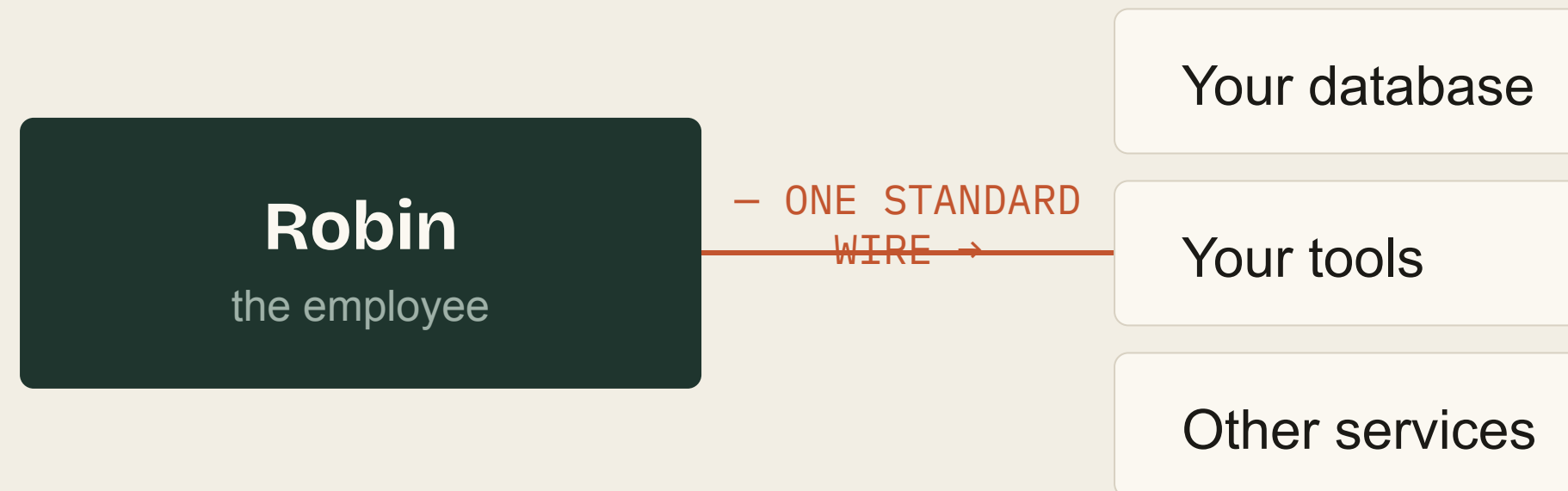
Customer: "the lid won't click shut"

Past case: "container doesn't seal" → **found**. Same problem, different words.

This is what lets Robin answer "we've seen this — here's what fixed it last time."

# The wire: how it reaches your real systems

Abilities and memory are useless if the employee can't safely touch your systems. The wire is one open, standard plug between them — think USB-C for software.



One standard means you can swap what's on either end without rewiring everything. Connect once; reuse everywhere.

# Two hands on the system, by design

A safety pattern worth understanding: the hand that builds the system and the hand that uses it day-to-day are deliberately different.

While building

## You and your agent — full access

Create the database, set things up, inspect anything. Wide powers, used by a human in the room.

While running

## Robin — narrow access only

Just the few specific things it needs: look up a customer, issue a refund. It can't reach beyond that.

The running employee never holds the master keys. It gets a small, purpose-built set — which is exactly what keeps it safe to let loose.

# Trust: how the business stays in control

An employee that acts on its own is only useful if you can trust it. Two simple disciplines make that real — a complete record, and a sign-off on the things that matter.

# Every action leaves a trace

The moment the employee does something, it writes down that it did — in the same breath, not as an afterthought. The record and the action are one and the same.

16:02:11	<code>looked_up</code>	customer #882, last 5 chats
16:02:14	<code>checked_policy</code>	refund window – within 30 days ✓
16:02:31	<code>refund_issued</code>	₹1,200 → order #4471 · approved by you

Anyone can replay exactly what happened, in order. That replayable record is what turns "the AI did something" into something a business can actually trust.

# Big actions wait for a human

Not every action deserves the same trust. Looking things up is harmless; moving money is not. So you draw a line — and the employee pauses at it.

Reads — run freely

## Look up a customer · find a past case

Nothing changes in the world. Let it go; no gate needed.

Writes — wait for sign-off

## Issue a refund · change a record

**Robin** stops and reports: *"waiting on your approval to refund ₹1,200."* Nothing moves until you say yes.

You decide where the line sits. The employee handles the routine and brings the consequential decisions to you.

# Putting it to work in your business

You know the parts. Now the practical questions: which job goes first, how you build it, and where it grows from there.

# Which jobs to hand over first

The best first hire is a job that's **repetitive**, **rule-based**, **leaves a record**, and is **low-risk** if it errs. Look for those four traits in any team.

## Support

### Triaging & answering tickets

Look up, reply, escalate the hard ones.

## Operations

### Order & status lookups

Where's my package, change an address.

## Finance

### Routine reconciliations

Match records, flag the exceptions.

## People / HR

### Answering policy questions

Leave, benefits, "how do I..." from the handbook.

Start narrow and boring. A first hire that quietly nails one job earns the trust to take on the next.

# How you actually build one

It looks like a conversation, because it is one. You describe the next small step; your agent plans it; you approve; it builds. Here's a real exchange:

**YOU** "Give Robin a tool to issue a refund — but make it wait for my sign-off before any money moves."

**YOUR AGENT** "Here's the plan: add an **issue\_refund** action, log it to the trace, and pause for your approval. Approve to build?"

Your real skill isn't typing — it's writing a clear brief and judging whether the result does what you meant.

PUTTING IT TO WORK

# Start with a fifteen-minute win

Don't build the whole employee in one weekend. Build the smallest version that actually works — then study why it worked, and grow from there.

---

THE FIRST WIN

One memory, one action:  
Robin saves a note and reads  
it back.

You see, with your own eyes, that it remembered.

Succeeding once, fast, beats planning forever. The architecture never changes — only how much of it you've built.

---

THEN IT GROWS

More skills, real records, the  
wire, the sign-off.

Same shape, one piece at a time. Each piece teaches you  
the next.

# It starts work on its own

A chatbot sits silent until you type. An employee is wired to the events of your business — it wakes itself when there's work, and no one has to kick it off.

A clock

## On a schedule

It runs at set times, whether or not anyone is around.

---

**Robin:** every morning at 7, flags overnight orders that didn't ship.

An event

## When something happens

A new ticket, a failed payment, a form submitted — the event itself starts it.

---

**Robin:** a refund request lands and Robin opens the case at once.

A handoff

## When another system calls

Another app — or another AI employee — hands it a task to pick up.

---

**Robin:** the store flags a VIP order, so Robin replies to them first.

Still on your terms: it wakes only for the triggers you set, and the sign-off on big actions still holds.

Autonomy inside the rules you wrote.

PUTTING IT TO WORK

# From one task to a full-time worker

TODAY

## On demand

Does one task when you ask.  
You're in the loop for everything.

NEXT

## Around the clock

Stays on, handling the routine  
while you sleep.

THEN

## Acts on its own

Picks up work when something  
happens — a new ticket, a  
schedule.

LATER

## Reaches you anywhere

Finds you on chat, email, or app  
when it needs a decision.

Same employee, same parts. What grows is how much you trust it to run without you — earned one reliable result at a time.

IN CLOSING

**What changes for you**  
**You stop being the one who does the**  
**work, and become the one who **directs****  
**and **reviews** it.**

Writing the brief replaces doing the task. Checking the result replaces building it by hand. That judgement — knowing what good looks like — is the skill that matters now, and it's one you already have.

---

THE WHOLE THING, IN ONE LINE

An AI employee is **abilities**,  
**memory**, and a **wire** — with a  
**trace of everything it does.**

ABILITIES

Skills it picks up

MEMORY

A source of truth

THE WIRE

A safe, standard plug

+ TRUST

A trace & a sign-off

Your first hire is one conversation  
away.

YOU DIRECT · YOUR AGENT BUILDS