

AI for a DIFC wealth firm, *running on a server in your own office*

The full system: the machine in the cabinet, the code that runs on it, the DIFC and DFSA rules it meets, the twelve jobs it does with an adviser signing each one, and the training that lets your own team run it. No client name, document or number ever reaches the internet.

0

CLIENT
RECORDS ON
THE INTERNET

12

WORKFLOWS,
ADVISER
SIGNS EACH
ONE

5

SERVICES ON
ONE OFFICE
SERVER

6 yrs

ENCRYPTED
RECORDS, TWO
DRIVES

4

TRAINING
TRACKS, ONE
PER ROLE

• 1 • TWO WAYS TO START

Setup, training, *or both*

Same system, same rules. The only difference is who builds it.

TRACK A • SETUP

I build it in your office

- ✓ Server installed, encrypted and cut off from the internet
- ✓ Twelve workflows tested on sample clients, then on yours
- ✓ AI policy, AI register and impact assessment written with your compliance officer
- ✓ Every login handed back at go-live, runbook and recorded walkthrough

Best for: firms with no in-house IT. Four phases, first pilot in week 3.

TRACK B • TRAINING

Your team builds and runs it

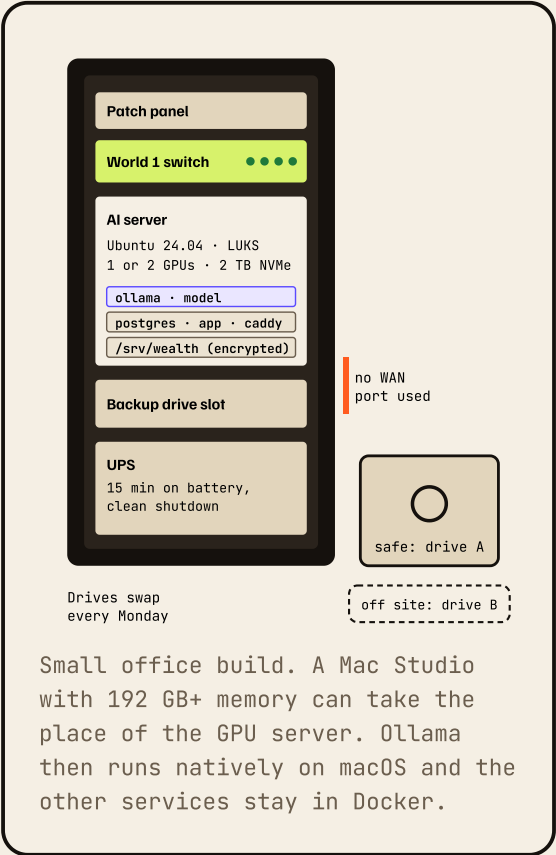
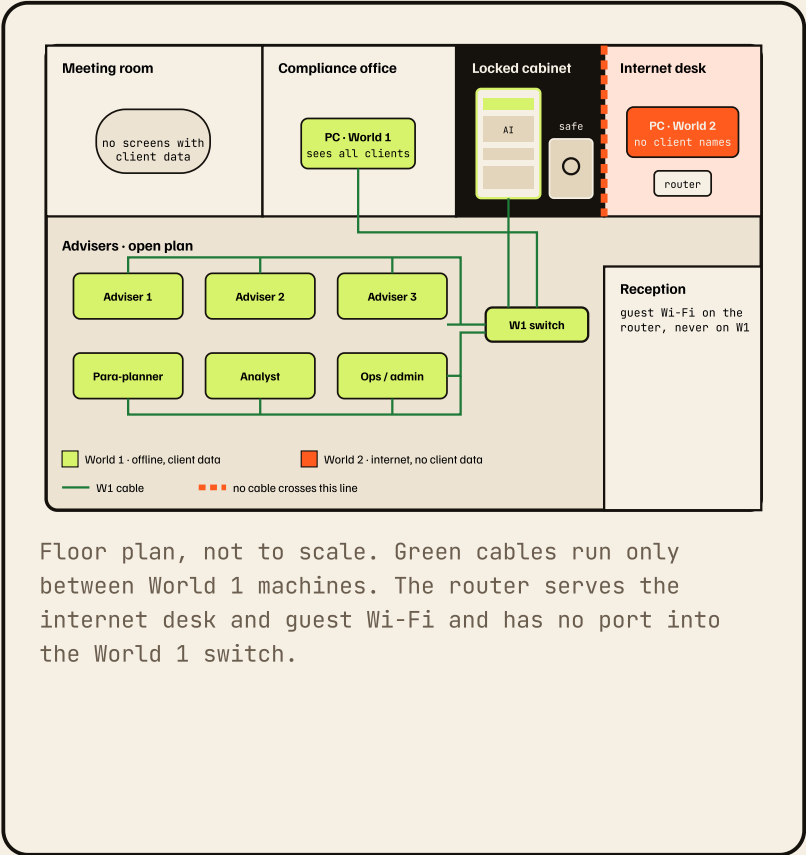
- ✓ Leadership: 90 minutes on what to approve and what to refuse
- ✓ Advisers: half a day on the desk, drafts and checking sources
- ✓ Compliance: half a day on the register and audit-log sampling
- ✓ IT: a full day on the server, backups and restore tests, using the kit below

Best for: firms with an IT person or managed provider. Taught at your DIFC office. [See the four tracks](#)

Most firms do both: Track A for the build, then Track B so the people who use it every day know how it works and where the limits sit. The DFSA expects the firm, not the supplier, to understand and own the system.

One cabinet, *two networks*

A five to ten person DIFC office needs one locked cabinet, one safe and one desk that stays on the normal internet. Nothing else changes for visitors or clients.

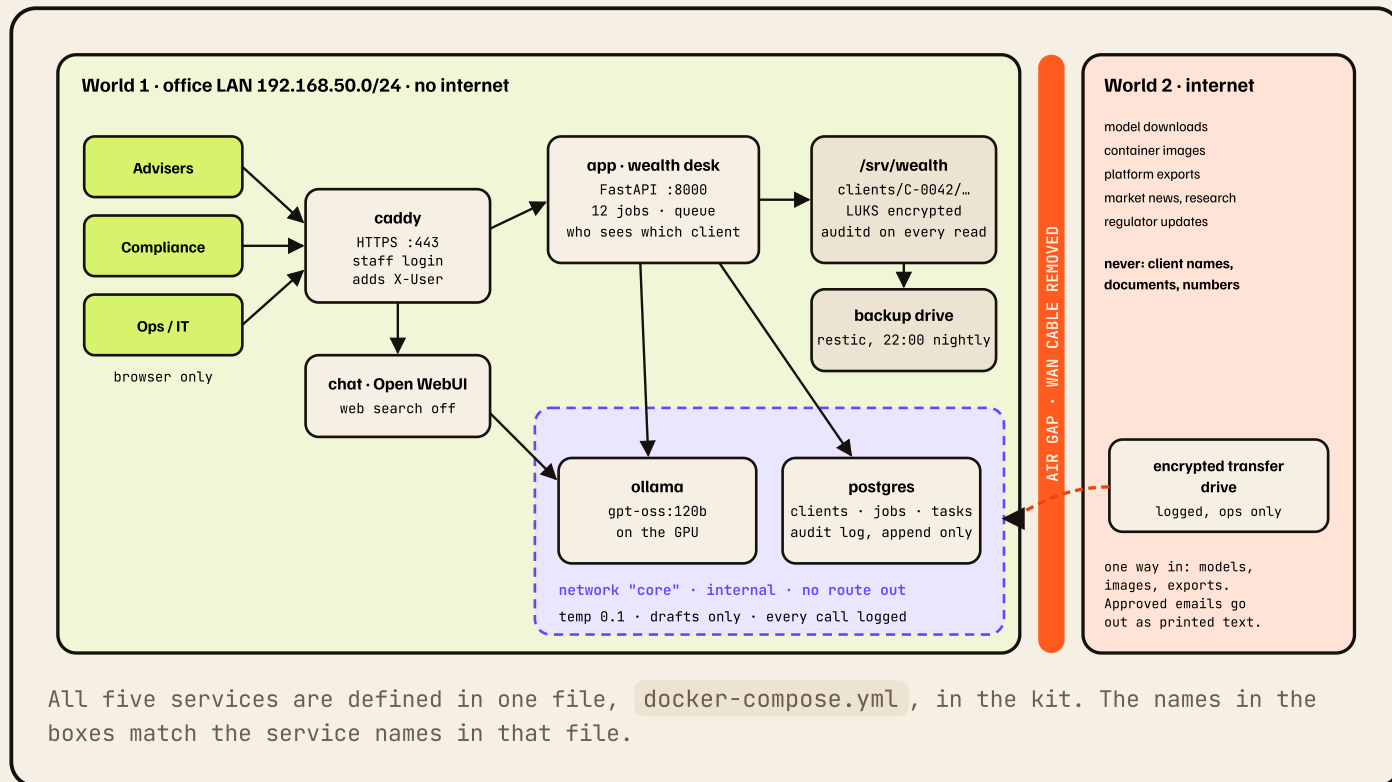


SIZE	MACHINE	FITS	MODEL THAT RUNS WELL	PLANNING RANGE
Small	Mac Studio, 192 GB+ unified memory, or a workstation with one 48 GB professional GPU	5 to 10 staff, documents and drafts	gpt-oss:120b or qwen3:32b	AED 20k to 50k
Medium	Rack server with one or two 80 GB GPUs, 256 GB RAM	10 to 40 staff, many documents a day	gpt-oss:120b with vLLM	AED 150k and up
Both	UPS, managed switch, two 4 TB encrypted USB drives, small safe	Every build	n/a	AED 4k to 8k

Model licences (Apache 2.0 for gpt-oss and Qwen3) reviewed once by counsel. Sizing guide: [Internal](#) · models: [Context Studios](#) · prices are planning ranges, Dubai retail, October 2026.

How a request *moves through the server*

Staff open `desk.office.lan` in a browser. Only the HTTPS gateway faces the office network. The model and the database sit on an internal Docker network with no route out, not even to the LAN.



● 4 · SAFEGUARDING THE DATA

Two worlds, *no link between them*

The one rule everything hangs on: **client data never touches the internet**. Anything with a client in it lives in World 1. Public work lives in World 2. Nothing crosses except on an approved encrypted drive.

WORLD 1 • OFFLINE, INSIDE THE
DIFC OFFICE

Client data and the AI that reads it

- **File server:** encrypted at rest, one folder per client (KYC, fact-find, suitability, portfolio, correspondence, reviews)
- **Database:** client register, pipeline stages, follow-up tracker, AI audit log
- **AI model:** open-weight, on the server in the cabinet. Downloaded once in World 2 and carried in
- **Backups:** two encrypted drives, one in the office safe, one off site, swapped weekly. Six-year retention
- **Access:** advisers see their own clients only, compliance sees all, every open and edit logged
- **Rules:** no cloud sync, no USB without approval, laptops encrypted, nothing to personal email

NO LINK

WORLD 2 • THE NORMAL INTERNET
DESK

Public work, no client names

- Prospect research, market news, regulator updates
- The prompts and skills on azizsaif.com/ai-wealth
- Platform exports downloaded here, carried to World 1 on the transfer drive
- Approved client emails typed or pasted on the mail desk from the signed draft
- Written rule in the AI policy: no client name, document or account number, ever

What the firm gives up offline, and the answer

- ✓ **No web research in World 1.** Done in World 2 with no client names.
- ✓ **No email or drive connectors in World 1.** Approved emails are copied to the normal mail desk for sending.
- ✓ **A slightly weaker model than the best cloud models.** For documents, drafts and summaries at a 5 to 10 person firm, it is enough.
- ✓ **Someone owns the machine.** Updates, backups and the drive swap: the firm's IT support, trained in Track B, or a monthly visit.

• 5 • THE RULES THIS MEETS

DIFC and DFSA, *line by line*

RULE	WHAT IT ASKS	WHAT THIS DESIGN DOES	WHERE IN THE BUILD	READ MORE
DIFC Data Protection Law No. 5 of 2020	The firm is the controller: lawful basis, security measures, data subject rights, breach reporting	Data never leaves the firm. Access logged. Retention set per client	01-setup-server.sh , retain_until column	DIFC Commissioner
DIFC DP Regulation 10 (autonomous and semi-autonomous systems)	Tell clients an AI system is used and name it. Keep a human in control. Keep records. High-risk uses may need an Autonomous Systems Officer and Commissioner approval	AI drafts, adviser signs. A named line in the client agreement. The audit log is the record. Onboarding assessed before go-live	jobs table status, ai_audit_log	Regulation text · Clyde & Co
DFSA GEN 5.3 systems and controls, outsourcing 5.3.21 and 5.3.22	Written agreement with any supplier, risk assessment, contingency and exit plan, firm stays responsible	Written scope and NDA, all logins handed back, runbook and Track B training so the firm runs it alone	README.md runbook	DFSA GEN module
DFSA record keeping	Client files, suitability reports and logs kept and retrievable. Six years is the period cited for governance records	Server plus two offline backups kept 72 months, restore tested quarterly	ops/backup.sh , ops/restore-test.sh	DFSA GEN module
DFSA AI Survey 2025 expectations	Written AI policy, clear accountability, an AI register. 26% of firms using AI in critical areas had no governance	Policy, register and named owner written in phase 1, before any AI runs	cLaude/CLAUDE.md , register template	DFSA survey

Read this first: these are summaries, not the rulebook. The firm's compliance officer confirms each line against the current DFSA Rulebook and DIFC Commissioner guidance before sign-off. That review is built into phase 1.

DOCUMENT 1

AI-use policy

What AI may do, what it may never do, who signs off, how clients are told, what happens on a breach.

DOCUMENT 2

AI register

One line per use: purpose, data touched, human owner, risk level, date reviewed. Each line maps to one key in app/jobs.py .

DOCUMENT 3

Impact assessment

For onboarding and KYC first, since that is where personal data is heaviest. Repeated for each new workflow.

Nine steps, *with the code*

Every file below is in the [server kit](#). Tap a file name to open the code. It is sample code: your IT support and compliance officer review it before live use.

1

DAY 1 • WORLD 1 SERVER

Install an encrypted server, then roles and folders

Ubuntu Server 24.04 LTS, full-disk encryption ticked in the installer. The script creates three roles, the folder tree, an idle lock and an audit watch on every client file.

▼ 01-setup-server.sh

bash

```
#!/usr/bin/env bash
# World 1 server setup. Ubuntu Server 24.04 LTS, run once as root.
# Disk encryption (LUKS) is chosen in the Ubuntu installer, before this script.
set -euo pipefail

# 1. Roles: advisers see their own clients, compliance sees all, ops runs the box
groupadd -f advisers; groupadd -f compliance; groupadd -f ops

# 2. Folder tree. One folder per client is created later by ops/new-client.sh
install -d -m 0750 -g compliance /srv/wealth/{clients,models,inbox,audit,exports}
install -d -m 0770 -g advisers /srv/wealth/inbox
chmod 1770 /srv/wealth/inbox          # sticky: staff cannot delete each other's
drops

# 3. Packages that work offline once installed
apt-get update
apt-get install -y docker.io docker-compose-v2 acl restic cryptsetup jq ufw
postgresql-client
systemctl enable --now docker

# 4. Auto-lock idle sessions and log every login
echo 'TMOUT=900; readonly TMOUT; export TMOUT' > /etc/profile.d/idle-lock.sh
systemctl enable --now auditd 2>/dev/null || apt-get install -y auditd
auditctl -w /srv/wealth/clients -p rwa -k client_files # every open and edit,
logged

# 5. Firewall: run 02-firewall.sh next, then unplug the WAN cable for good
echo "Done. Next: ./02-firewall.sh"
```

2

DAY 1 • WORLD 1 SERVER

Block the internet, prove it, unplug it

Deny all outgoing traffic, then block containers in the `DOCKER-USER` chain, since Docker skips `ufw`. The last line must print `OK: no internet`. Screenshot it for the compliance file, then pull the WAN cable.

▼ 02-firewall.sh

bash

```
#!/usr/bin/env bash
# Second layer behind the physical air gap. The office LAN is 192.168.50.0/24.
set -euo pipefail
LAN=192.168.50.0/24

ufw --force reset
ufw default deny incoming
ufw default deny outgoing
ufw allow from $LAN to any port 443 proto tcp # the dashboard, over HTTPS
ufw allow from $LAN to any port 22 proto tcp # ops only, key login
ufw allow out to $LAN
ufw --force enable

# Docker writes its own rules and skips ufw. Block container traffic leaving the
# LAN.
iptables -I DOCKER-USER -d $LAN -j RETURN
iptables -I DOCKER-USER 2 -d 172.16.0.0/12 -j RETURN
iptables -A DOCKER-USER -j DROP
apt-get install -y iptables-persistent && netfilter-persistent save

# Proof for the compliance file: this must FAIL
curl -s --max-time 5 https://example.com && echo "WARNING: internet reachable" ||
echo "OK: no internet"
```

3

DAY 2 • WORLD 2 DESK, THEN THE TRANSFER DRIVE

Carry the model and images in

On the internet desk: `ollama pull gpt-oss:120b`, `docker compose build`, then `docker save` every image to the encrypted transfer drive. On the server: `docker load` and copy the model folder to `/srv/wealth/models`. Ops logs the drive in and out.

▼ .env.example

env

```
# Copy to .env and change every value. Never commit .env.
POSTGRES_PASSWORD=change-me-long-random
MODEL=gpt-oss:120b # small office: gpt-oss:20b or qwen3:32b
OLLAMA_URL=http://ollama:11434
# Mac Studio: run Ollama natively, then use
# OLLAMA_URL=http://host.docker.internal:11434
```



Start five services with one file

`docker compose up -d`. The model and database live on the `core` network marked `internal: true`. Only Caddy publishes a port, and only to the office LAN.

```
# World 1 stack. Start with: docker compose up -d
name: wealth-desk

networks:
  core:          # model and database. No route out, not even to the LAN
    internal: true
  lan:          # only Caddy publishes a port to the office network
    driver: bridge

services:
  ollama:        # the AI model, on the firm's own GPU
    image: ollama/ollama:latest
    networks: [core]
    volumes: ["/srv/wealth/models:/root/.ollama"]
    environment: ["OLLAMA_KEEP_ALIVE=24h"]
    deploy:
      resources:
        reservations:
          devices: [{driver: nvidia, count: all, capabilities: ["gpu"]}]]
    restart: unless-stopped

  db:            # client register, review queue, audit log
    image: postgres:16
    networks: [core]
    environment:
      POSTGRES_DB: wealth
      POSTGRES_USER: wealth
      POSTGRES_PASSWORD: ${POSTGRES_PASSWORD}
    volumes:
      - /srv/wealth/db:/var/lib/postgresql/data
      - ./db/schema.sql:/docker-entrypoint-initdb.d/01-schema.sql:ro
    restart: unless-stopped

  app:           # the workflow dashboard (drop, draft, review, approve, file)
    build: ./app
    networks: [core, lan]
    environment:
      DATABASE_URL: postgresql://wealth:${POSTGRES_PASSWORD}@db:5432/wealth
      OLLAMA_URL: ${OLLAMA_URL}
      MODEL: ${MODEL}
      CLIENT_ROOT: /srv/wealth/clients
    volumes:
      - /srv/wealth/clients:/srv/wealth/clients
      - /srv/wealth/inbox:/srv/wealth/inbox
    depends_on: [db, ollama]
    restart: unless-stopped

  chat:          # Open WebUI: a private ChatGPT-style screen for staff
    image: ghcr.io/open-webui/open-webui:main
    networks: [core, lan]
```

```

environment:
  OLLAMA_BASE_URL: ${OLLAMA_URL}
  ENABLE_SIGNUP: "false"
  ENABLE_WEB_SEARCH: "false"
  OFFLINE_MODE: "true"
volumes: ["/srv/wealth/webui:/app/backend/data"]
restart: unless-stopped

caddy:          # HTTPS on the office LAN, staff login, passes the user name
on
  image: caddy:2
  networks: [lan]
  ports: ["443:443"]
  volumes: ["/Caddyfile:/etc/caddy/Caddyfile:ro", "/srv/wealth/caddy:/data"]
  restart: unless-stopped

```

▼ Caddyfile

caddy

```

# Internal certificate. Install Caddy's root cert on office PCs once.
{
  local_certs
}

desk.office.lan {
  basic_auth {
    # caddy hash-password --plaintext 'their-password'
    sara    $2a$14$REPLACE_WITH_HASH
    omar    $2a$14$REPLACE_WITH_HASH
    comply $2a$14$REPLACE_WITH_HASH
  }
  reverse_proxy app:8000 {
    header_up X-User {http.auth.user.id}
  }
}

chat.office.lan {
  reverse_proxy chat:8080
}

```



The register, the queue and the audit log

Loaded on first start. The audit log refuses updates and deletes, so nobody can tidy it later. Each client gets a `retain_until` date six years from opening.

```
-- Who may see what
CREATE TABLE staff (
  username  text PRIMARY KEY,
  full_name text NOT NULL,
  role      text NOT NULL CHECK (role IN ('adviser','compliance','ops'))
);

CREATE TABLE clients (
  client_id text PRIMARY KEY,          -- e.g. C-0042, never the name
  adviser   text NOT NULL REFERENCES staff,
  stage     text NOT NULL DEFAULT 'enquiry', -- enquiry, kyc, suitability, active,
  review
  risk_level text,
  opened_on  date NOT NULL DEFAULT current_date,
  retain_until date GENERATED ALWAYS AS (opened_on + interval '6 years') STORED
);

-- Every AI job waits here until a person approves it
CREATE TABLE jobs (
  id          bigserial PRIMARY KEY,
  job_type    text NOT NULL,
  client_id   text REFERENCES clients,
  source_file text NOT NULL,
  draft       text,
  status      text NOT NULL DEFAULT 'drafted' CHECK (status IN
('drafted','approved','rejected')),
  created_by  text NOT NULL REFERENCES staff,
  approved_by text REFERENCES staff,
  created_at  timestampz NOT NULL DEFAULT now(),
  decided_at  timestampz
);

CREATE TABLE tasks (
  id          bigserial PRIMARY KEY,          -- staff follow-up tracker
  client_id   text REFERENCES clients,
  owner       text NOT NULL REFERENCES staff,
  action      text NOT NULL,
  due_on      date NOT NULL,
  done        boolean NOT NULL DEFAULT false
);

-- The record the regulator reads. Append only.
CREATE TABLE ai_audit_log (
  id          bigserial PRIMARY KEY,
  at          timestampz NOT NULL DEFAULT now(),
  username    text NOT NULL,
  action      text NOT NULL,          -- draft, approve, reject, view
  job_id      bigint,
  model       text,
  prompt_sha  text,
```

```
    output_sha text,  
    note      text  
);  
REVOKE UPDATE, DELETE ON ai_audit_log FROM PUBLIC;  
CREATE RULE no_update AS ON UPDATE TO ai_audit_log DO INSTEAD NOTHING;  
CREATE RULE no_delete AS ON DELETE TO ai_audit_log DO INSTEAD NOTHING;
```



Drop, draft, review, approve, file

About a hundred lines of Python. An adviser drops a file, the model drafts with page references, the draft waits in the queue, and only an approve call writes it into the client folder. Every step is hashed into the audit log. Advisers see their own clients. Compliance sees all.

```

"""Wealth desk: drop a file → AI drafts → review queue → adviser approves →
filed + logged."""
import hashlib, os, pathlib, httpx, psycpg
from fastapi import FastAPI, Header, HTTPException, UploadFile, Form
from pypdf import PdfReader
from jobs import HOUSE_RULES, JOBS

DB = os.environ["DATABASE_URL"]
OLLAMA = os.environ["OLLAMA_URL"]
MODEL = os.environ["MODEL"]
ROOT = pathlib.Path(os.environ["CLIENT_ROOT"])
app = FastAPI(title="Wealth desk (World 1)")

def sha(t: str) → str: return hashlib.sha256(t.encode()).hexdigest()

def log(cur, user, action, job_id=None, prompt="", output="", note=""):
    cur.execute("INSERT INTO
ai_audit_log(username,action,job_id,model,prompt_sha,output_sha,note)"
               " VALUES (%s,%s,%s,%s,%s,%s,%s)",
               (user, action, job_id, MODEL, sha(prompt), sha(output), note))

def role_of(cur, user):
    row = cur.execute("SELECT role FROM staff WHERE username=%s",
(user,)).fetchone()
    if not row: raise HTTPException(403, "unknown user")
    return row[0]

def may_see(cur, user, client_id):
    if role_of(cur, user) == "compliance": return True
    row = cur.execute("SELECT adviser FROM clients WHERE client_id=%s",
(client_id,)).fetchone()
    return bool(row) and row[0] == user

def read_text(f: UploadFile) → str:
    if f.filename.lower().endswith(".pdf"):
        pages = PdfReader(f.file).pages
        return "\n".join(f"[p.{i+1}]\n{p.extract_text() or ''}" for i, p in
enumerate(pages))
    return "[p.1]\n" + f.file.read().decode("utf-8", "replace")

@app.post("/jobs")
def new_job(job_type: str = Form(...), client_id: str = Form(...), file: UploadFile
= None,
           x_user: str = Header(...)):
    if job_type not in JOBS: raise HTTPException(400, "unknown job")
    with psycpg.connect(DB) as con, con.cursor() as cur:
        if not may_see(cur, x_user, client_id): raise HTTPException(403, "not your
client")
        prompt = f"{HOUSE_RULES}\n\nTASK:
{JOBS[job_type]}\n\nDOCUMENT:\n{read_text(file)}"

```

```

        r = httpx.post(f"{OLLAMA}/api/generate", timeout=600,
                      json={"model": MODEL, "prompt": prompt, "stream": False,
                            "options": {"temperature": 0.1}})
    draft = r.json()["response"]
    job_id = cur.execute(
        "INSERT INTO jobs(job_type,client_id,source_file,draft,created_by)"
        " VALUES (%s,%s,%s,%s,%s) RETURNING id",
        (job_type, client_id, file.filename, draft, x_user)).fetchone()[0]
    log(cur, x_user, "draft", job_id, prompt, draft)
    return {"job": job_id, "status": "drafted", "draft": draft}

@app.get("/queue")
def queue(x_user: str = Header(...)):
    with psycopg.connect(DB) as con, con.cursor() as cur:
        sql = ("SELECT j.id,j.job_type,j.client_id,j.created_by,j.created_at FROM
jobs j "
              "JOIN clients c USING(client_id) WHERE j.status='drafted'")
        if role_of(cur, x_user) != "compliance":
            sql += " AND c.adviser=%s"
            rows = cur.execute(sql, (x_user,)).fetchall()
        else:
            rows = cur.execute(sql).fetchall()
        log(cur, x_user, "view", note="queue")
        return [dict(zip(["id", "type", "client", "by", "at"], r)) for r in rows]

@app.post("/jobs/{job_id}/{decision}")
def decide(job_id: int, decision: str, final_text: str = Form(""), x_user: str =
Header(...)):
    if decision not in ("approve", "reject"): raise HTTPException(400)
    with psycopg.connect(DB) as con, con.cursor() as cur:
        job = cur.execute("SELECT client_id,job_type,draft FROM jobs WHERE id=%s
AND status='drafted'",
                          (job_id,)).fetchone()
        if not job or not may_see(cur, x_user, job[0]): raise HTTPException(404)
        text = final_text or job[2]
        cur.execute("UPDATE jobs SET status=%s, approved_by=%s, decided_at=now()
WHERE id=%s",
                    (decision + "d", x_user, job_id))
        if decision == "approve": # filed into the client's own
folder
            out = ROOT / job[0] / job[1] / f"{job_id}-approved-by-{x_user}.md"
            out.parent.mkdir(parents=True, exist_ok=True)
            out.write_text(text)
        log(cur, x_user, decision, job_id, output=text)
        return {"job": job_id, "status": decision + "d"}

```

```
# One instruction per workflow. The firm's compliance officer owns this file.
HOUSE_RULES = """You work for a DIFC-regulated wealth firm. Rules:
- Draft only. A named adviser approves everything you write.
- Never recommend a product, fund or security.
- Every figure carries its source as [p.N] from the page markers.
- If a field is not in the document, write MISSING. Never guess.
- Plain English. No promises about returns."""

JOBS = {
    "kyc": "Extract: full name, date of birth, nationality, passport number and expiry, "
          "residential address, source of wealth, source of funds, employer, annual income. "
          "Return one line per field as FIELD: value [p.N]. List every MISSING field at the end.",
    "meeting_prep": "Write a one-page meeting brief: who, goals on file, holdings vs model, "
                    "open actions, three questions to ask. Max 300 words.",
    "file_note": "Turn these meeting notes into a file note: attendees, what was discussed, "
                 "what the client asked for, decisions, actions with owner and due date.",
    "suitability": "Draft a suitability report from the fact-find and risk profile: client "
                   "objectives, risk level and why, how the model portfolio fits, risks, costs. "
                   "Leave the recommendation section as [ADVISER TO WRITE].",
    "drift": "Compare holdings to the model weights. List every line outside +/-5 points "
            "with current %, target %, and the gap. No trade suggestions.",
    "research": "Summarise this report in one page. Every figure with [p.N]. End with "
               "'What this does not say' in three bullets.",
    "client_update": "Draft a short personal update email for this client from their numbers "
                     "and open actions. Warm, factual, under 180 words. No forecasts.",
    "screening": "Apply the house screening rules below to each security. For each: PASS or "
                 "FAIL and the rule that decided it.",
}
```

```
FROM python:3.12-slim
WORKDIR /app
# Build this image on the World 2 desk, then: docker save | load on the server
```

▼ ops/new-client.sh

bash

```
#!/usr/bin/env bash
# Usage: sudo ./new-client.sh C-0042 sara
# Creates the client folder. Only the named adviser and compliance get in.
set -euo pipefail
ID=$1; ADVISER=$2; D=/srv/wealth/clients/$ID
install -d -m 0700 "$D"/{kyc,fact-
find,suitability,portfolio,correspondence,reviews}
setfacl -R -m u:$ADVISER:rwX -m g:compliance:rX "$D"
setfacl -R -d -m u:$ADVISER:rwX -d -m g:compliance:rX "$D"
psql "$DATABASE_URL" -c "INSERT INTO clients(client_id,adviser) VALUES
('$ID','$ADVISER')"
echo "Client $ID opened for $ADVISER"
```

Nightly encrypted backup, quarterly restore test

Two drives swap every Monday: one in the safe, one off site. 72 monthly snapshots cover six years. Once a quarter compliance picks a random client and ops restores it while both watch.

▼ ops/backup.sh

bash

```
#!/usr/bin/env bash
# Nightly at 22:00 via cron. Two encrypted drives (A and B) swap every Monday:
# one in the office safe, one off-site.
set -euo pipefail
DRIVE=$(ls -d /media/backup-[AB] | head -1)      # whichever drive is plugged
in
export RESTIC_REPOSITORY=$DRIVE/restic
export RESTIC_PASSWORD_FILE=/root/.restic-pass    # also sealed in the safe on
paper

docker compose -f /opt/wealth-desk/docker-compose.yml exec -T db \
  pg_dump -U wealth wealth > /srv/wealth/exports/db-$(date +%F).sql

restic backup /srv/wealth --exclude /srv/wealth/models --tag nightly
restic forget --keep-daily 14 --keep-weekly 52 --keep-monthly 72 --prune    # 72
months = 6 years
restic check --read-data-subset=5%
echo "$(date -Is) backup ok to $DRIVE" >> /srv/wealth/audit/backup.log
```

▼ ops/restore-test.sh

bash

```
#!/usr/bin/env bash
# Quarterly. Compliance picks a random client, ops restores it, both sign the log.
set -euo pipefail
CLIENT=${1:?client id}
T=/tmp/restore-test-$(date +%F); rm -rf "$T"
restic restore latest --target "$T" --include "/srv/wealth/clients/$CLIENT"
find "$T" -type f | wc -l
echo "$(date -Is) restore test $CLIENT ok, files: $(find "$T" -type f | wc -l)" >>
/srv/wealth/audit/restore.log
rm -rf "$T"
```

Lock the build tool

Claude Code is used to build and test the workflows on sample data. Managed settings deny web access and client folders, and the hook logs every tool call. Users cannot switch these off. Details in the next section.

▼ **claude/managed-settings.json**

json

```
{
  "permissions": {
    "defaultMode": "default",
    "disableBypassPermissionsMode": "disable",
    "deny": [
      "WebFetch",
      "WebSearch",
      "Bash(curl:*)",
      "Bash(wget:*)",
      "Bash(scp:*)",
      "Bash(ssh:*)",
      "Read(/srv/wealth/clients/**)",
      "Edit(/srv/wealth/clients/**)"
    ]
  },
  "hooks": {
    "PreToolUse": [
      { "matcher": "*",
        "hooks": [ { "type": "command", "command": "/opt/wealth-desk/claude/audit-hook.sh" } ] }
    ]
  }
}
```

▼ **claude/audit-hook.sh**

bash

```
#!/usr/bin/env bash
# Runs before every Claude Code tool call. Logs it, and blocks client paths.
# Exit code 2 = blocked, and the reason goes back to Claude.
IN=$(cat)
TOOL=$(jq -r '.tool_name' <<<"$IN")
ARGS=$(jq -c '.tool_input' <<<"$IN")
echo "$(date -Is) $USER $TOOL $ARGS" >> /srv/wealth/audit/claude-code.log
if grep -q '/srv/wealth/clients' <<<"$ARGS"; then
  echo "Blocked: client folders are off limits to the build tool. Use sample data."
  >&2
  exit 2
fi
exit 0
```

```
# Firm rules for Claude Code (build machine)
```

- You build and test workflows. You never see live client data.
- Test files live in ./samples and use made-up clients only (C-9001 to C-9099).
- Never write code that sends data outside 192.168.50.0/24.
- Every new workflow needs: an entry in app/jobs.py, a line in the AI register, and a test on three sample files before the compliance officer sees it.
- Drafts only. Nothing in this system sends an email or places a trade.

● 7 · THE SAFEGUARDS, ENFORCED BY SETTINGS

What Claude Code *locks down*

Claude Code builds and tests the workflows. Its controls are plain settings files the compliance officer can read, not promises in a sales deck.

ASK BEFORE ACTING

Every file write, command or email is shown first and needs a person's approval. Bypass mode is disabled in managed settings, so no user can turn approvals off.

DENY LISTS

Settings block tools outright:

WebFetch , WebSearch , curl ,
wget , scp , ssh , and any read or
edit under
/srv/wealth/clients .

HOOKS THAT LOG AND BLOCK

A hook runs before every tool call, writes who, what and when to `claude-code.log` , and stops any call that names a client path.

MANAGED SETTINGS

The file lives in `/etc/claude-code/` on Linux or `/Library/Application Support/ClaudeCode/` on macOS, owned by root. User settings cannot override it.

RULES IN PLAIN ENGLISH

`CLAUDE.md` carries the firm's rules: sample clients only, no outbound code, every workflow tested on three files and added to the register.

NO TRAINING ON YOUR DATA

Under Anthropic's commercial terms, API and enterprise data is not used to train models. Zero-data-retention is available for enterprise accounts.

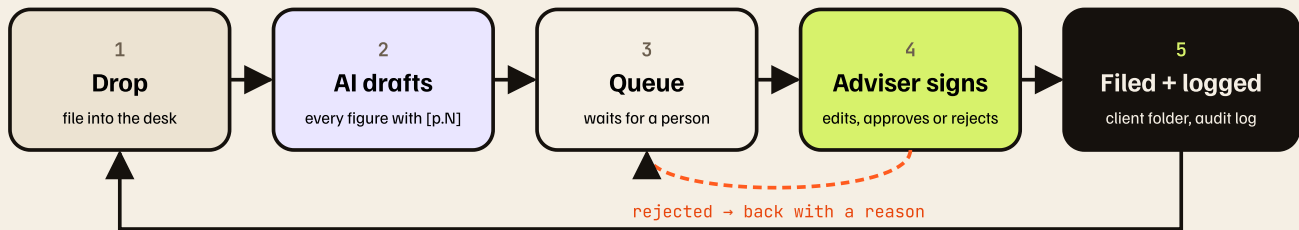
Honest limit: Claude Code needs a connection to a model, so it runs on a World 2 build machine with sample data. Live client documents are read only by the open-weight model inside World 1. A firm that accepts an enterprise cloud account with the controls above can choose to run more in the cloud. That is the firm's decision, made in the impact assessment.

Documentation: [Claude Code settings and permissions](#) · [hooks](#) · [Anthropic commercial terms](#)

● 8 · TWELVE WORKFLOWS, ONE LOOP

How each job *flows*

Every job runs the same five steps. Only the input and the person who signs change.

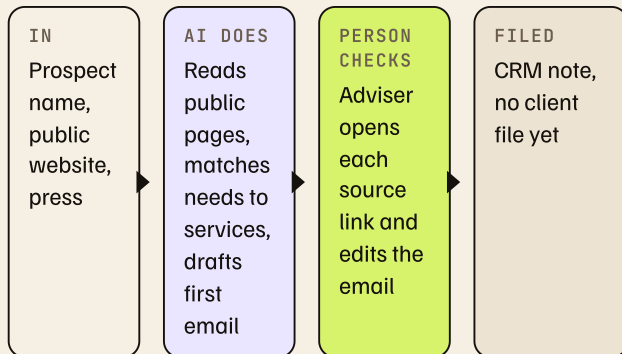


next job for the same client starts from the approved file

☐ input ☐ AI does ☒ person checks ☐ output and where it is filed

1 Prospect brief

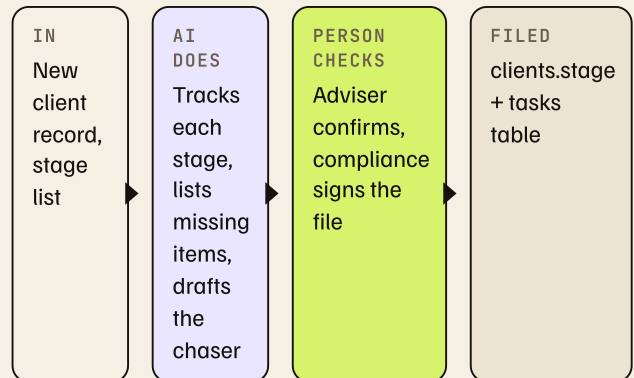
WORLD 2



job key World 2 only

2 Onboarding pipeline

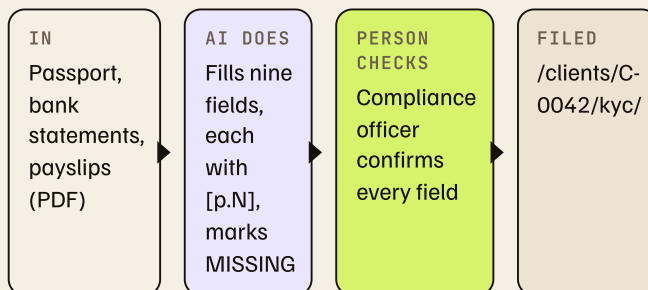
WORLD 1



job key tasks today 3 to 4 h per client

3 Fact-find and KYC extraction

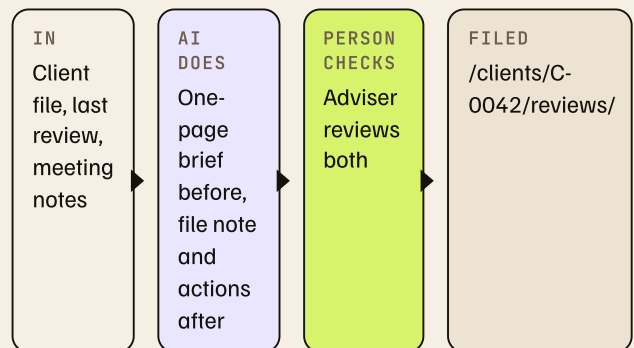
WORLD 1



job key kyc today 3 to 4 h per client

4 Meeting prep and file note

WORLD 1



job key meeting_prep, file_note
today 30 to 45 min before, 30 after

5

Suitability report

WORLD 1

IN
Fact-
find,
risk
profile,
model
portfolio

AI DOES
First draft,
recommendation
left blank

PERSON CHECKS
Adviser writes
the
recommendation
and signs

FILED
/clients/
0042/su

job key suitability

6

Portfolio drift check

WORLD 1

IN
Holdings
export,
model
weights

AI DOES
Lists
every
line
outside
+/-5
points

PERSON CHECKS
Adviser
decides
any
rebalance

FILED
/clients/C-
0042/portfolio/

job key drift today 45 min per quarterly check

7

Equity and bond screening

WORLD 1

IN
Security
list,
house
screening
rules

AI DOES
PASS or
FAIL
per
security
with
the rule

PERSON CHECKS
Analyst
signs
the
shortlist

FILED
/research/screens/

job key screening

8

Research notes

WORLD 1

IN
40-page
research
report
(PDF)

AI DOES
One
page,
every
figure
with
[p.N]

PERSON CHECKS
Analyst
clicks
references,
adds
house
view

FILED
/research/notes/

job key research today An afternoon per report

9

News watch

WORLD 2

IN
Public
news,
regulator
updates

AI DOES
Matches
stories to
client
segments,
rates
impact

PERSON CHECKS
Adviser
reads
the
primary
source

FILED
Segment
list, no
client
names

job key World 2 only

10

Client update emails

WORLD 1

IN
Client
numbers,
open
actions

AI DOES
Personal
draft
under
180
words,
no
forecasts

PERSON CHECKS
Adviser
approves,
sends
from the
mail desk

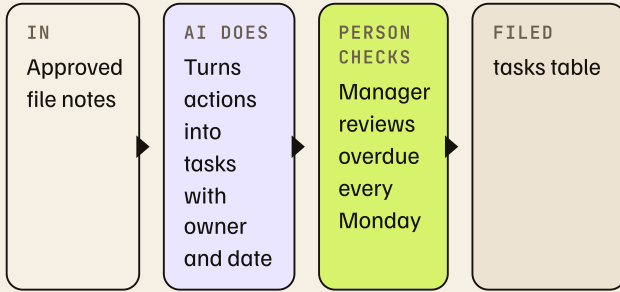
FILED
/clients/C-
0042/correspondenc

job key client_update
today 90 to 120 min per note

11

Staff follow-up tracker

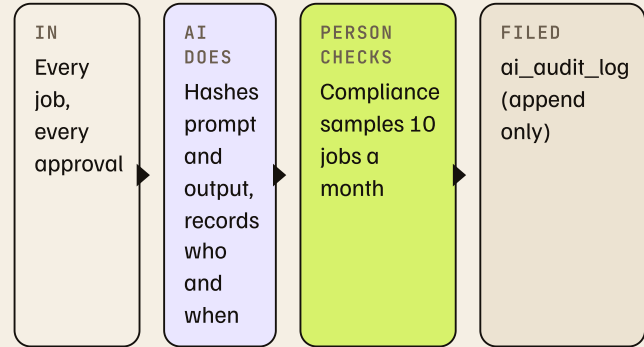
WORLD 1

job key `file_note` → tasks

12

Audit log and verification

WORLD 1

job key `automatic`

World 1 = offline office network. World 2 = internet desk, no client data. Job keys match `app/jobs.py`. "Today" times are industry ranges reported by [Automaton \(May 2026\)](#), a vendor, and are replaced by the firm's own pilot figures. Interactive version: azizsaif.com/wealth-demo

● 9 • CLEAN GUIDELINES FOR STAFF

The rules on *one wall*

Print this section and pin it next to every World 1 desk. It is the short version of the AI-use policy.

Always

- ✓ Use client IDs (C-0042) in file names, never names
- ✓ Click every [p.N] reference before you approve
- ✓ Fix MISSING fields with the client, never with a guess
- ✓ Write your own recommendation section in suitability reports
- ✓ Reject a weak draft with a reason so the prompt improves
- ✓ Lock your screen when you stand up (Windows key + L)

Never

- ✗ Paste client data into any website, app or phone chat
- ✗ Use a personal USB stick or personal email for client files
- ✗ Send an AI draft to a client before an adviser approves it
- ✗ Let AI pick a product, fund or security
- ✗ Plug a World 1 machine into the router or guest Wi-Fi
- ✗ Share your desk login, even with the IT support person

Operations calendar

DAILY

Advisers

- Clear your review queue
- Overdue tasks first

WEEKLY ·
MONDAY

Ops

- Swap backup drives A and B
- Check `backup.log` shows 7 nights

MONTHLY

Compliance

- Sample 10 approved jobs
- Read the audit-log summary
- Update the AI register

QUARTERLY

Ops +
compliance

- Restore test on a random client
- Model and patch update via transfer drive

YEARLY

Partners

- Review policy and impact assessments
- Delete files past `retain_until`

If something goes wrong

WHAT HAPPENED	FIRST 10 MINUTES	WHO
Client data pasted into a website	Screenshot, close the tab, tell compliance. Do not delete anything. Compliance assesses a breach notice to the DIFC Commissioner	Staff member, then compliance
A draft went to a client unapproved	Call the client, correct in writing, log it in the audit table as a note	Adviser, then compliance
Backup drive lost	Drives are encrypted, so record it. Rotate the restic password, take a fresh backup on a new drive	Ops
Server will not start	Work from paper files. Ops follows the runbook. Restore to spare hardware from the safe drive if needed	Ops

● 10 · TRAINING · TRACK B

Four tracks, *one per role*

Taught at your DIFC office on the firm's own server, using made-up sample clients. Each person leaves able to do their part without calling anyone.

90 MINUTES

Partners and leadership

- ✓ What the twelve jobs do and do not do
- ✓ Regulation 10 duties in plain words
- ✓ Reading the monthly audit summary
- ✓ Approving the policy and register

HALF DAY • 4 HOURS

Advisers and para-planners

- ✓ Dropping files and choosing the job
- ✓ Checking [p.N] and MISSING fields
- ✓ Editing, approving and rejecting well
- ✓ Private chat for drafting, the never-list

HALF DAY • 4 HOURS

Compliance officer

- ✓ Writing the AI register and impact assessment
- ✓ Sampling the audit log with three queries
- ✓ Client disclosure wording
- ✓ Breach steps and the restore test

FULL DAY • 7 HOURS

IT and ops

- ✓ Server build from the kit, step 1 to 9
- ✓ Firewall proof and the transfer drive
- ✓ Adding staff, clients and a new job
- ✓ Backup, drive swap and restore

What compliance runs each month

▼ monthly-sample.sql

sql

```
-- 1. Who did what with AI in the last 30 days
SELECT username, action, count(*)
FROM ai_audit_log
WHERE at > now() - interval '30 days'
GROUP BY 1, 2 ORDER BY 1;

-- 2. Ten random approved jobs to read in full
SELECT id, job_type, client_id,
created_by, approved_by, decided_at
FROM jobs WHERE status = 'approved'
ORDER BY random() LIMIT 10;

-- 3. Anything approved by the same person who created it
SELECT id, job_type, client_id,
created_by FROM jobs
WHERE status = 'approved' AND
created_by = approved_by;
```

How a training day runs

TIME BLOCK

09:00 The two worlds, with the cabinet open in front of the group

09:45 Hands-on: each person runs one job on a sample client

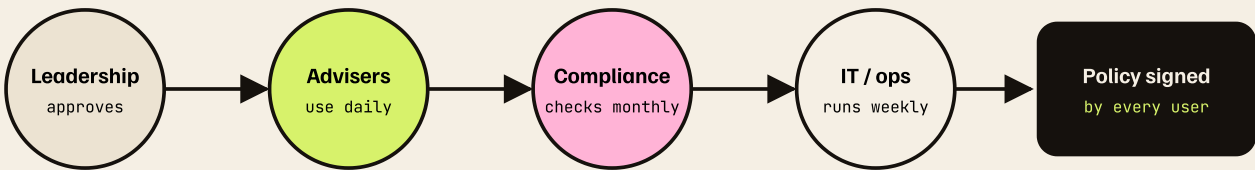
11:00 Find the planted mistake: drafts with one wrong figure each

12:00 The never-list, and what to do in the first 10 minutes

13:00 Role track: advisers, compliance or IT

15:30 Each person signs the AI-use policy

Groups of 4 to 12. Fees and dates: [AI training in Dubai](#) • DIFC firms: [AI automation for DIFC](#)



90 min • 4 h • 4 h • 7 h • one office, sample clients only

• 11 • IMPLEMENTATION PLAN

Four phases, *first pilot in week 3*

1

WEEKS 1 TO 2 •
FOUNDATIONS

- Build steps 1 to 9, on sample clients
- AI-use policy, AI register, impact assessment
- Compliance officer reviews the rules table
- Outsourcing agreement and NDA signed

2

WEEKS 3 TO 4 •
PILOT

- One job: meeting prep, one adviser
- Runs beside the old way
- Hours logged daily
- Go or no-go on real numbers

3

MONTH 2 •
ONBOARDING

- KYC extraction and onboarding pipeline
- Follow-up tracker
- Adviser and compliance training
- First audit-log sample by compliance

4

MONTHS 3 TO 4 •
FULL DESK

- Drift, screening, research, client updates, suitability
- IT training and first restore test
- Runbook and recorded walkthrough handed over
- Compliance review, then decide what's next

Handed over at the end

Working desk on the firm's own server, with every login and password returned

AI-use policy, AI register, impact assessment, audit-log procedure

Runbook, the server kit, a recorded walkthrough and four training tracks delivered

• 12 • QUESTIONS DIFC FIRMS ASK

Straight *answers*

Can a DIFC wealth firm use AI without sending client data to the cloud?

Yes. An open-weight model such as gpt-oss-120b runs on a server inside the office with the internet cable removed. Client documents are read only on that server. The model, software and updates are downloaded on a separate internet desk and carried in on an encrypted drive.

What hardware does a small DIFC wealth firm need for offline AI?

For five to ten staff, one Mac Studio with 192 GB or more of memory, or a workstation with one 48 GB professional GPU, plus a UPS, a switch, two encrypted backup drives and a small safe. Planning range is AED 20,000 to 50,000 for the machine. Ten to forty staff need a rack server with one or two 80 GB GPUs, from AED 150,000.

Which DIFC and DFSA rules apply to AI at a wealth firm?

The DIFC Data Protection Law No. 5 of 2020, Regulation 10 on autonomous and semi-autonomous systems, DFSA GEN 5.3 on systems and controls and outsourcing, DFSA record keeping, and the governance expectations in the DFSA AI Survey 2025. The firm's compliance officer confirms each against the current rulebook before sign-off.

Do we have to tell clients we use AI?

Under DIFC Data Protection Regulation 10, clients are told an AI system is used and what it does. This build adds a named line to the client agreement. The AI drafts and an adviser approves, so a person stays in control of every output.

How long does the setup take?

Four phases. Server, policy and register in weeks 1 and 2, a one-adviser pilot in weeks 3 and 4, onboarding and KYC in month 2, and the full twelve-job desk in months 3 and 4.

Can our own staff run the system after setup?

Yes. Training comes in four tracks taught at the firm's DIFC office: 90 minutes for partners, half a day for advisers, half a day for the compliance officer, and a full day for IT on the server, backups and restore tests. The server kit and runbook are handed over.

Where does Claude Code fit if client data stays offline?

Claude Code builds and tests the workflows on a separate build machine using made-up sample clients. Managed settings deny web access and any read of client folders, disable bypass mode, and a hook logs every tool call. Live client documents are read only by the model inside the office.

How are records kept for six years?

Nightly encrypted backups with restic to two drives that swap every Monday, one in the office safe and one off site. 72 monthly snapshots are kept. Every quarter compliance picks a random client and ops restores it to prove the backup works.

● NEXT STEP

A 30-minute call, then the 15-minute demo with your team

Say setup, training or both. NDA before any document is shared. Phase 1 can start the week after.

[Book a 30-min call](#)[WhatsApp](#)[Email](#)

Prepared by Aziz Saif, AI automation consultant, Dubai. Related: [AI automation for DIFC firms](#) · [five wealth jobs and prompts](#) · [15-minute demo](#) · [case study](#) · [AI training in Dubai](#) · [one-page proposal PDF](#). Time savings are planning estimates from the sources linked and are replaced by the firm's own pilot figures. The code is sample code, reviewed by the firm's IT and compliance before live use. Nothing here is legal advice or investment advice. Sources checked 4 to 7 October 2026.