**LIFE‑SUPPORT — CHECKPOINT REPORT (MINI & GATEKEEPING)**

*Date:* 2025‑08‑18  
*Owner:* Almamoun  
*Scope:* Mini (agent) side — DB flags, upsert/notify API, run scripts. Includes key rules & open items so **we can pause/resume safely.**

**Short answer: Yes — everything we created is in place at these paths:**

**E:\life-support-mini\api\stop\_mini\_api.bat ← stops the API**

**E:\life-support-mini\open\_mini\_ui.bat ← (optional) starts API then opens the page**

**C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\Start Mini API.lnk ← auto-start on login**

* **DB: E:\life-support-mini\db\mini.db**
* **SQL: E:\life-support-mini\db\alter\_upsert.sql**
* **API: E:\life-support-mini\api\server.js**
* **Scripts: E:\life-support-mini\api\start\_mini\_api.bat, E:\life-support-mini\api\stop\_mini\_api.bat**
* **Autostart: C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\Start** Mini API.lnk

Note: the **/gatekeeping/status** route was appended to server.js; it’s live **only after a restart** of the API.

### One-step verify (PowerShell)

Run this single line to confirm file existence + sizes:

$paths = @('E:\life-support-mini\db\mini.db','E:\life-support-mini\db\alter\_upsert.sql','E:\life-support-mini\api\server.js','E:\life-support-mini\api\start\_mini\_api.bat','E:\life-support-mini\api\stop\_mini\_api.bat',"$env:APPDATA\Microsoft\Windows\Start Menu\Programs\Startup\Start Mini API.lnk"); $paths | % { $i=Get-Item -LiteralPath $\_ -ErrorAction SilentlyContinue; [pscustomobject]@{Path=$\_; Exists=[bool]$i; Size=if($i){$i.Length}; LastWrite=if($i){$i.LastWriteTime}} } | Format-Table -AutoSize

**1) WHAT’S WORKING NOW (VERIFIED TODAY)**

* **SQLite schema in Mini DB** supports tri‑state gatekeeping:
  + gate\_status\_lu values: **0=pending**, **1=permitted**, **2=held\_elsewhere**
  + gatekeeping(entity\_type, entity\_guid, status, source, note, updated\_at) with PK (entity\_type, entity\_guid)
  + event\_outbox table + triggers:
    - on **INSERT** and on **UPDATE(status)** → enqueue gatekeeping.changed rows
* **Mini API (HTTP, port 3001)** running and tested:
  + GET /health → returns { ok, db, python }
  + POST /gatekeeping/upsert → UPSERT + auto‑notify via DB trigger
  + GET /event-outbox/peek → shows newest events (tested)
  + GET /event-outbox/drain?limit=N → returns undelivered events **and marks delivered** (tested)
* **Start/Stop & Autostart**
  + start\_mini\_api.bat (starts API with Mini DB path)
  + stop\_mini\_api.bat (kills Node on port 3001 / by script path)
  + Startup shortcut created (auto‑launch on login)

**Smoke tests performed:**

* Upserted GUID → **status: permitted (1)** ✓
* Updated same GUID → **status: held\_elsewhere (2)** ✓
* Outbox created 3 events; **drain** marked them delivered ✓

**2) FILES & FOLDERS (MINI)**

* **API code:** E:\life-support-mini\api\server.js  
  (Python bridge; no native sqlite3 build needed.)
* **DB:** E:\life-support-mini\db\mini.db
* **SQL migration script:** E:\life-support-mini\db\alter\_upsert.sql
* **Scripts:**
  + E:\life-support-mini\api\start\_mini\_api.bat
  + E:\life-support-mini\api\stop\_mini\_api.bat
* **Autostart link:**  
  C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\Start Mini API.lnk
* **Legacy Prospecting mini (separate, HTTPS 3443)** still present alongside new API (kept intact):
  + serves prospects.html & prospects.json

**3) API CONTRACT (MINI)**

* **POST /gatekeeping/upsert**
  + **Body** { entity\_guid, status, entity\_type='prospect', source='mini', note }
  + status: number **0|1|2** or strings \*\*'pending'|'permitted'|'held\_elsewhere'`
  + **Effect**: UPSERT gatekeeping + trigger inserts into event\_outbox
  + **Return**: { ok:true, entity\_type, entity\_guid, status, source, note }
* **GET /event-outbox/peek** → { ok, rows:[...] } (last 10)
* **GET /event-outbox/drain?limit=50`** → returns undelivered rows oldest‑first **and sets delivered\_at**
* **GET /health** → { ok, db, python }
* *(Planned)* **GET /gatekeeping/status?guid=...** — code appended; requires API restart to load.

**4) RUNBOOK — QUICK COMMANDS (POWERSHELL)**

* **Health**

Invoke-RestMethod -Method Get -Uri 'http://127.0.0.1:3001/health'

* **Upsert** (example permitted)
* $g=[guid]::NewGuid().ToString();

Invoke-RestMethod -Method Post -Uri 'http://127.0.0.1:3001/gatekeeping/upsert' -ContentType 'application/json' -Body (@{entity\_guid=$g; status='permitted'; source='mini'; note='test'} | ConvertTo-Json)

* **Drain**

Invoke-RestMethod -Method Get -Uri 'http://127.0.0.1:3001/event-outbox/drain?limit=100'

* **Start/Stop**
* E:\life-support-mini\api\start\_mini\_api.bat

E:\life-support-mini\api\stop\_mini\_api.bat

**5) DECISIONS & RULES (CONFIRMED)**

* **Operational model:** *Main → Mini → Main* via encrypted .lsx bundles; Main is merge‑only (**MASTER‑1**) and system of record.  
  Minis see assigned slices; **NOASSIGN→NODB**; **Export is copy‑only**; **KILL‑ONEWAY** supported.
* **Data placement (PBI/PLCD):**  
  PBI allowed on Main+Mini; PLCD default on Minis; Sponsors may include PLCD on Main only when ordered/admin‑scoped.
* **Prospecting UX:** Single interface; admin merges duplicates; Mini cannot interfere with another Mini’s active record; city/company keyed under GAT (GAT→City→Company triad).
* **Gatekeeping:** Mini cannot work a record unless **permitted** flag returns true from Main. Interim states: **pending** (awaiting), **held\_elsewhere** (another assignee).
* **Promotion:** Two prospect types (payers vs non‑payers) with referral loop; rule: *“If it appears in sponsor webpage then it is legitimate.”*
* **Playbook Notes:** Main owns versioned playbook; Minis auto‑update cached copy; must\_say flag.

**6) WHAT’S PARTIALLY DONE / WHERE IT LIVES**

* **Sponsor record spec:** defined (GAT ID, contract signed/expiry, feature toggles like doctor list update) — lives in design notes and admin tool description. Implementation in DB/UI pending.
* **Participant/onboarding flow:** draft script & Arabic copy exist; UI wiring to backend not finalized here.
* **Floating gift icon (entry to Life‑Support):** demo scripts/pages were explored earlier; current repo has legacy prospecting mini (HTTPS 3443). Button → should call new **/gatekeeping/upsert** when creating/claiming a record; not yet wired.

**7) OPEN ITEMS / BACKLOG (PRIORITIZED)**

1. **Status endpoint online** — Restart API to load /gatekeeping/status route that was appended.
2. **UI wiring** — From the floating gift / mini UI, call /gatekeeping/upsert on “Start Work / Claim”.
3. **Main consumer** — Implement Main‑side poller to hit event-outbox/drain, process, and ACK (already marks delivered on drain).
4. **.LSX integration** — Add gatekeeping & outbox handling into export/import merge logic.
5. **Prospecting mini consolidation** — Unify HTTPS (3443) UI with the new HTTP (3001) API or proxy one to the other.
6. **Portable package** — Build mini\_api.exe with pkg (or embed Python); produce /dist folder with start/stop; optional Inno Setup for Setup.exe.
7. **Sponsor/Admin tools** — Implement gat-admin.exe behaviors (define GAT, expiry, feature toggles; doctor list permission).
8. **Security** — Tighten local CORS as needed; consider token for drain endpoint if exposed beyond localhost.

**8) NEXT 3 MICRO‑STEPS (WHEN WE RESUME)**

1. **Restart API** to enable /gatekeeping/status (then test with last GUID).
2. **Add one line in UI**: on create/claim, call the upsert endpoint; show badge **pending/permitted/held**.
3. **Write Main poller** (tiny script) to call /event-outbox/drain and log applied actions.

**9) NOTES / RISKS**

* Node native sqlite3 build failed on this machine; current Python‑bridge solution is stable and acceptable for Mini.
* Legacy prospecting server (3443) and new API (3001) are separate; avoid confusion by standardizing on one start script.

**10) APPENDIX — SQL OBJECTS**

* gate\_status\_lu(code,name) → seed (0,pending),(1,permitted),(2,held\_elsewhere)
* gatekeeping(entity\_type, entity\_guid, status, source, note, updated\_at)
* event\_outbox(id, topic, entity\_type, entity\_guid, status, payload, created\_at, delivered\_at)
* Triggers: trg\_gatekeeping\_notify\_insert, trg\_gatekeeping\_notify\_update