## CHECKLIST

## 1 Data-model tweaks for check-lists

| **Field** | **Type** | **Purpose** |
| --- | --- | --- |
| description | string | Task wording (“Establish command.”) |
| critical | boolean | true = must be complete before the overall checklist can be signed off; renders with alert icon & red outline. |
| completed | boolean | User-toggled checkbox. |
| completedBy | user ID | Auto-filled from auth claim when box ticked. |
| completedAt | ISO datetime | Timestamp of completion. |

### Digital signature block (per checklist form)

"signature": {

"signed": true,

"signedById": "user-uuid",

"signedByName": "A. Garcia",

"signedAt": "2025-07-08T14:37:05Z",

"method": "pin" // future-proof for “draw” or CAC

}Signing is disabled until ***all critical:true items are completed.***

## 3 Widget & API flow

1. **First time any user clicks “Sign”**
   * App checks /user/{id}/pin → 404 → prompts “Create 6-digit PIN” (enter twice).
   * POST /user/{id}/pin → server stores bcrypt(pin, salt).
2. **Normal signing**
   * User types PIN.
   * App POSTs /auth/verify-pin (userId, pin).
   * If OK, server returns JWT claim pinVerified=true (120 s TTL).
   * App writes signature block, computes HMAC, saves new FormVersion.
3. **Changing PIN** (profile page)
   * PUT /user/{id}/pin with currentPin + newPin. Needs currentPin match.
4. **Security controls**
   * 5 bad attempts ⇒ account lock 10 min (rate-limited at server).
   * PIN must be exactly 6 numeric digits.
   * bcrypt cost factor 12.

## 4 Database migration snippet

sql

CopyEdit

CREATE TABLE UserPin (

userId UUID PRIMARY KEY REFERENCES "User"(id),

pinHash TEXT NOT NULL,

createdAt TIMESTAMPTZ DEFAULT now(),

updatedAt TIMESTAMPTZ DEFAULT now()

);

(Existing users will have no row; they’ll be prompted to set one on first signature.)

## 5 SDK helper

ts

CopyEdit

// pseudo-code

async function ensurePinVerified(userId: UUID): Promise<void> {

if (tokenHas('pinVerified') && !tokenExpired()) return;

const pin = await promptPin();

await api.verifyPin(userId, pin);

## 2 Draft JSON-schema stubs

Below are **very slim starter schemas** (just a couple of items each) so you can see the pattern. We’ll flesh them out once you give the full text for every item.

### 2.1 IC Startup (v1)

json

CopyEdit

{

"$id": "checklist-ic-startup-v1",

"title": "Incident Commander – Startup",

"type": "object",

"properties": {

"items": {

"type": "array",

"items": {

"type": "object",

"properties": {

"description": { "type": "string" },

"critical": { "type": "boolean", "default": true },

"completed": { "type": "boolean", "default": false },

"completedBy": { "type": "string", "nullable": true },

"completedAt": { "type": "string", "format": "date-time", "nullable": true }

},

"required": ["description", "critical", "completed"]

},

"default": [

{ "description": "Establish Incident Command Post (ICP)", "critical": true },

{ "description": "Name the Incident", "critical": true },

{ "description": "Open 204 task board", "critical": false }

]

},

"signature": { "$ref": "#/definitions/signature" }

},

"definitions": {

"signature": {

"type": "object",

"properties": {

"signed": { "type": "boolean", "default": false },

"signedBy": { "type": "string", "nullable": true },

"signedAt": { "type": "string", "format": "date-time", "nullable": true }

}

}

}

}

### 2.2 Safety Officer – Ongoing (v1)

Example items only; mark those you treat as critical.

json

CopyEdit

"default": [

{ "description": "Conduct scene safety sweep each hour", "critical": true },

{ "description": "Monitor weather & heat index", "critical": false },

{ "description": "Log all hazards in 214", "critical": true }

]

### 2.3 PIO – First Press Release (v1)

json

CopyEdit

"default": [

{ "description": "Draft first release using template", "critical": true },

{ "description": "Get IC approval of wording", "critical": true },

{ "description": "Post to county website & social media", "critical": false }

]

## 3 UI/UX behavior

* **Checklist screen**
  + Critical items show a red border until completed.
  + Progress bar counts critical items separately (2 / 5 critical done).
* **Signature action**
  + “Sign & Complete” button disabled until all critical items checked.
  + Once signed, checklist becomes read-only and exports with signature block.
* **Dashboard roll-up**
  + IC sees a widget: “Safety Officer Ongoing – 80 % critical complete”.
  + Overdue critical items (<time threshold>) trigger push alert.

## 4 Next steps & what I need from you

| **Step** | **Owner** | **When** |
| --- | --- | --- |
| **A. Supply full item lists** for the three checklists (plain text or existing docs). | **You** | As soon as convenient |
| **B. Confirm any extra metadata** (priority levels beyond critical/non-critical, additional signers, etc.). | **You** | — |
| **C. Implement critical-aware checklist renderer** (tap UX, red outlines, progress logic). | **Me** | immediately after A |
| **D. Add digital-signature field type** (name, date/time auto-stamp). | **Me** | parallel with C |
| **E. Build PDF template** (simple two-column checklist layout + signature line). | **Me** | after C/D |

Once you send the actual checklist items, I’ll drop them into the schemas, wire up the renderer, and show you a quick mock so you can judge the look & feel before we commit.

### Incident & Event Lifecycle — detailed design draft

(for review and editing before we freeze the schema)

#### 1 . Entity & state model

| **Entity** | **Core fields** | **Key states** | **Notes** |
| --- | --- | --- | --- |
| **Incident** (or **Planned Event**) | id (GUID) · name · type (fire, search, parade…) · createdBy · geoBounds | **Draft** → **Active** → **Closed** → **Archived** | • “Draft” lets staff pre-stage an event without publishing. • “Archived” = read-only ZIP on object store. |
| **Operational Period** | incidentId · number · start/end (UTC) · objectivesDraft | **Open** → **Locked** | Period is Locked when IC signs IAP; all attached forms become version-stamped. |
| **Org Assignment** (ICS-203 row) | periodId · positionCode · personId · radioCallsign | — | Drives RBAC claims automatically. |
| **Objective** (ICS-202) | periodId · text · priority (1-5) · status | **Proposed** → **Approved** → **Carried-Fwd / Complete** |  |
| **FormInstance** | periodId · formType (201, 204, custom…) · version · payload (JSON) · locked bool | Draft → Final | Each save bumps a minor-rev; Final when period is Locked. |

Why separate Operational Period? It’s the “time-slice” that snapshots objectives, assignments, and forms for the IAP.

#### 2 . Typical workflow (happy path)

| **Step** | **Actor** | **Screen / Form** | **Data actions** |
| --- | --- | --- | --- |
| 1. **Create** incident/event | Planning Section or IC | “New Incident / Event” wizard (name, type, map bounding box) | Inserts **Incident** row (Draft) + **OP-1** row (Open). |
| 2. Populate **ICS-201 or Event Template** | IC/Plans | ICS-201 form viewer | Writes **FormInstance(201)**. [training.fema.gov](https://training.fema.gov/emiweb/is/icsresource/assets/ics%20forms/ics%20form%20201%2C%20incident%20briefing%20%28v3%29.pdf?utm_source=chatgpt.com) |
| 3. Build **Org Chart** | IC/Plans | Drag-and-drop 203 grid | Writes **OrgAssignment** rows. |
| 4. Add **Objectives** | Plans | Objectives tab (202 boxes 4-6) | Objective rows (Proposed). |
| 5. Create **Assignments (204)** | Ops | Task board | **FormInstance(204)** per division. |
| 6. **Lock period** & publish IAP | IC | “Lock & Publish” dialog | – Sets **OP-1 Locked**. – All linked FormInstances marked Final. – Auto-generates IAP PDF booklet. |
| 7. Open next period | Plans | “Start Period 2” | Clones carry-forward objectives; Org chart optionally rolled over. |
| 8. End-incident closeout | IC | Close Incident | Sets Incident→Closed; triggers demob checks. |
| 9. Archive | Records admin | Archive button | Bundles JSON+PDF ZIP; Incident→Archived. |

Edge-/cloud sync pushes every state change; field devices work offline and merge when re-connected.

#### 3 . Permissions model (auto-derived)

| **Role claim** | **Create/Edit** | **View only** |
| --- | --- | --- |
| **IC** | All incident & period metadata, lock/unlock | — |
| **PlansSec** | Org chart, objectives, forms (201-209), open/close periods | After lock |
| **OpsSec** | Tasks/204, resource status | All after lock |
| Other command staff | Their position checklist/forms | Read current IAP |
| Viewer | — | IAP & sit-rep PDF downloads |

When a user is assigned to “Safety Officer” in the 203 for the current Period, they automatically receive the **SafetyOfficer** role claim until the period closes.

#### 4 . UI wire-flow (textual)

pgsql

CopyEdit

[Incident List] → tap incident

├─► [Summary] (name, map, status chips)

│ ├─ “Start Period” (if none open)

│ └─ “Close Incident”

├─► [Periods] tab

│ └─ list cards: OP-1 (Locked) • OP-2 (Open)

├─► [Org Chart] tab (ICS-203)

├─► [Objectives / 202] tab

├─► [Forms] tab (grid of all forms incl. custom)

└─► [IAP Exports] tab

Period badge colors: **Blue** Open, **Grey** Locked. Lock icon on the action bar once period is sealed.

#### 5 . API sketch (REST/gRPC)

bash

CopyEdit

GET /incidents?status=active

POST /incidents (creates Draft)

PATCH /incidents/{id} (rename, close, archive)

POST /incidents/{id}/periods (start next)

PATCH /periods/{pid} (lock/ unlock)

GET /periods/{pid}/org

PUT /periods/{pid}/org (bulk 203 update)

GET /periods/{pid}/forms?type=204

PUT /forms/{fid} (save draft)

POST /forms/{fid}/finalize (mark Final)

POST /periods/{pid}/export/iap (returns PDF zip)

All endpoints accept If-Offline-Queue: true header—SDK stores request locally until connectivity.

#### 6 . Edge cases & rules

* **Edits after lock:** Only IC can unlock; system keeps audit trail (unlockReason, user, timestamp).
* **Incident reopen:** Allowed only if it hasn’t been archived; reopens last period or forces a new one.
* **Overlap guard:** Cannot start a new Period whose start time precedes current period end.
* **Planned events recurrence:** Optional template flag so a parade you run yearly can be cloned.

#### 7 . MVP slice (suggested)

| **Must-have for first field test** | **Can slide to Phase 2** |
| --- | --- |
| Incident create/close/archive | Recurring event templates |
| OP create/lock | Period unlock audit log UI |
| ICS-201, 202, 203 core screens | Multi-incident merge |
| IAP booklet export | Automatic carry-forward rules tuning |
| Org-based RBAC | Public read-only portal |

**Updated Operational-Period & IAP flow**

| **Phase** | **What’s editable?** | **How the app versions your changes** |
| --- | --- | --- |
| **Period Open** *(0600 → …)* | - All ICS forms (201, 202, 204, 205, etc.) - Objectives - Org chart | Every save bumps an **incremental revision number** on that form (e.g., 204-DIV A v3). The latest revision is what shows in dashboards. |
| **Publish IAP (any time)** | Plans clicks **“Publish snapshot”**. | The system renders a *PDF snapshot* of every current form revision, bundles them, and stamps the set with a **snapshot ID + timestamp**. *Nothing locks*—you can still edit afterwards. |
| **Continue edits** | Teams keep updating forms/tasks. | New saves create **v4, v5, …**. Snapshots never change. |
| **Close Period** *(≈ 1830)* | Plans/IC clicks **“Close Period”**. | Period state flips to **Closed**. Forms lock, checklists sign, 214 rolls up. Incomplete tasks/objectives carry forward to the new period’s draft. |

### UI tweaks

* **“Lock Period” button becomes “Close Period.”**  
  Only available at end-of-shift; this is what freezes data.
* **New “Publish IAP Snapshot” button** in the Period toolbar.  
  Shows last publish time (e.g., “Last snapshot: 1037 EDT”).
* **Form headers** show “v3 (latest), v2, v1” with a dropdown to view / export any prior revision.

### Audit & integrity

* Snapshots are immutable PDF files stored in the object store (/iap-snapshots/{periodId}/{timestamp}.pdf).
* Revision history for each form is kept in the FormInstanceVersion table (who, when, diff hash).
* Closing a period automatically publishes one final snapshot **and** locks the forms.

### Benefits of this approach

* ICs/Ops can keep tweaking assignments or radio plan right up until briefing time.
* You still get legally defensible, time-stamped IAP copies.
* Field devices offline at snapshot time just keep editing; when they sync, they become the new latest revision (v n+1) without touching earlier snapshots.

### Module 2 — ****Planning Section****

A detailed design proposal you can tweak before we freeze the data model and start wiring UI.

## 2.1 Objectives & Strategy (ICS-202 core)

| **Topic** | **Design notes** |
| --- | --- |
| **Entity** | Objective (id, periodId, text, priority 1-5, status proposed / approved / complete / carried-forward, createdBy, createdAt, closedAt) |
| **Workflow** | 1️⃣ Draft objectives anytime in the period. 2️⃣ IC (or Plans) toggles **Approved**—they become part of the next IAP snapshot. 3️⃣ Ops or IC marks **Complete** as the period progresses. 4️⃣ When you Close Period, any **Approved + incomplete** objectives auto-roll into period + 1 as Carried-Forward. |
| **UI** | • Kanban-style board (Proposed → Approved → Complete). • Drag handle for priority order. • Badge shows linked tasks (e.g., “DIV A: 2/5 complete”). |
| **Snapshot rule** | At Publish IAP Snapshot time we freeze the **latest text** of each Approved objective into the PDF. Status can still move to Complete later. |
| **Aspect** | Implementation |
| **Priority field** | priority =1‒5 (integer). Table ObjectivePriorityEnum seeded with 1–Critical, 2–High, 3–Moderate, 4–Low, 5–Monitor so you still get labels. Later you can swap the enum table for a Life/Incident/Property/Env. scale without touching code. |
| **Schema change** | No enum constraint in JSON Schema—just "minimum":1,"maximum":5. |
| **UI tweak** | Dropdown shows numbers plus the label (e.g., “2 – High”) to future-proof a wording change. |

## 2.2 Forms Engine (standard + custom)

| **Feature** | **Details** | |
| --- | --- | --- |
| **Schema registry** | • Table FormType stores JSON-Schema + metadata. • Standard forms (201-232) pre-loaded; custom forms (checklists, local templates) added via the Custom Form Manager. | |
| **Revision model** | Table FormVersion (formId, version, payloadJSON, authorId, stamp, signatureBlock). Every save → version++ (v1, v2…); final bool when period closes. | |
| **Renderer** | Reads schema → dynamic widget tree (text/number/date, checklist, signature, map embed, file attach). | |
| **Validation** | JSON-Schema rules + extra logic (e.g., box 7 must equal period end). | |
| **PDF adapter** | For each formType, entry templateRef points to AcroForm PDF or DOCX. On export: payloadJSON → template → flattened PDF (via pdf-lib / iText). | |
| **Digital signatures** | Widget writes { signed: true, signedBy, signedAt } into payload; PDF overlay prints signature line. | |
| **Offline** | Whole form payload lives in device SQLite; diff-attributable merge (last-writer per JSON path). | |
| **Permissions** | Default: author + Plans + IC can edit; others read-only unless role matches requiredRole field in FormType. | |
| **Signature widget** | | *Typed name + PIN*: on first use the user sets a 6-digit PIN. Payload stores: { signed: true, signedByName, signedById, signedAt, pinHash (bcrypt) }. PDF overlay prints name+timestamp; PIN never printed. |

|  |  |
| --- | --- |
| **Agency-custom form order** | Table FormSequenceProfile (agencyId, defaultSequenceJSON). FEMA default pre-loaded; admins can reorder via drag-list UI and save a profile. Snapshot builder reads profile in priority: *incident.agencyId → default*. |

|  |  |
| --- | --- |
| **Revision dropdown** | Shows **v1, v2, … (latest)**; revision used in last snapshot is flagged with a 📄 icon. |

## 2.3 IAP Builder & Booklet (snapshot engine)

| **Step** | **Behind-the-scenes action** | | |
| --- | --- | --- | --- |
| Hit **Publish IAP Snapshot** | ① For current period, collect latest revision of every form where includeInIAP == true (flag stored in FormType). ② Render each to PDF stream. ③ Concatenate in configured order (201 → 202 → 203 → 204s → custom). ④ Save bundle at /iap-snapshots/{periodId}/{ts}.pdf and add Snapshot row (id, periodId, ts, authorId, formVersionMap). | | |
| Booklet layout | Options: • “Single-sided w/ TOC” • “2-up booklet”. Renderer uses pdf-lib to insert page numbers & table of contents. | | |
| Distribution | • Auto-e-mail list of “Plans + Command Staff” (address book). • Push notification links to snapshot. | | |
| Re-publish | Each snapshot is immutable; re-publishing in the same period creates /iap-snapshots/{periodId}/{ts2}.pdf. | | |
| Close Period | Auto-creates one **final** snapshot if the last one is > 15 min old, then locks forms. | | |
| **Auto-snapshot cadence** | | Every **4 hours** on the hour (00, 04, 08, 12, 16, 20 local). | Admin can set *Off*, **N hours**, or specific cron string in *Incident Settings → Governance*. |

|  |  |  |
| --- | --- | --- |
| **Manual button** | *Publish IAP Snapshot* still available anytime. |  |

|  |  |  |
| --- | --- | --- |
| **Snapshot metadata** | Snapshot row gains autoGenerated (bool) so reports can separate manual vs. scheduled. |  |

|  |  |  |
| --- | --- | --- |
| **Alerts** | Push/e-mail sent on every snapshot (**manual or auto**) to “Command + Plans” group. |  |

## 2.4 Situation / Intel Board

| **Capability** | **Implementation** |
| --- | --- |
| **Sit-Rep form (ICS-209)** | Standard form in Forms Engine, revisionable like others. |
| **Attachments** | Any file (photo, PDF, KML, weather briefing) uploaded to FileBlob, tagged sitreps, version-stamped. |
| **Weather widget** | Pull hourly NWS/NOAA JSON; cache snapshot of forecast into 209 each time you publish IAP. |
| **Map overlays** | Accept GPX/KML/GeoJSON; stored in PostGIS, toggle layers on map. |
| **Intel chat** | Channel scoped to incident (not period); messages optionally mark as “Intel” so they auto-copy into 209 narrative field. |
| **Dashboard cards** | – Current perimeter (acres, % contained). – Incident complexity level. – Active weather warnings (heat, lightning). |
| **Snapshot ties** | When you publish an IAP snapshot, the system copies the **latest 209 content** and embeds thumbnails of attachments (e.g., perimeter map image) in the PDF appendix. |

## 2.5 Integration with other modules

| **From / To** | **Data flow** |
| --- | --- |
| **Objectives ↔ Task Board (Module 3)** | When Ops creates a Task they can link it to an Objective ID; completion % shows on Objective card. |
| **Forms Engine ↔ Custom Form Manager (Module 7)** | Manager writes new FormType; Forms Engine auto-exposes it under “+ New Form.” |
| **IAP Snapshots ↔ Export & Archive (Module 7)** | ArchiveIncident() bundles all snapshot PDFs + JSON data into final ZIP. |
| **Intel Board ↔ Mapping Module (6)** | Uploaded KML overlays appear as map layers; trackers shown as heat-map for Sit-Rep metrics. |

### Module 3 — ****Operations Section****

A complete design draft for the Task & Assignment engine, field-reporting loop, and live dashboards. Tweak anything you like before we freeze the data model.

| **Decision you gave** | **How it changes the design** |
| --- | --- |
| **1. Status varies by task type** | Each *Task Type* owns its **own status flow** object, so Search tasks can show *En Route → Searching → Refitting* while Logistics tasks might be *Requested → In Progress → Delivered*. |
| **2. Priority is numeric** | priority now **1 (high) … 5 (low)**; labels in the UI (“1 – Critical”). |
| **3. No overdue rule (for now)** | Dashboard widget removed; table column left nullable so we can add SLA later. |
| **4. Photo uploads only** | File picker limited to JPEG/PNG; video MIME types rejected server-side. |
| **5. GPS every 60 s** | Mobile SDK sends a beacon **once per minute** while a task is *inside any “active” status* for its type. |

## 3.1 Task / Assignment Board (ICS-204 “guts”)

| **Concept** | **Details** |
| --- | --- |
| **Entity** | Task (id, periodId, divisionCode, title, description, objectiveId?, geoType (point / line / polygon / none), geoData (GeoJSON), priority (A–D), status, createdBy, createdAt, updatedAt) |
| **Status FSM** | **Planned → Briefed → In-Progress → Blocked / Needs-Assist → Complete → Demobbed** Rules: • Can’t skip ahead (UI disables illegal jumps). • **Ops Chief** can force-jump with audit trail. |
| **Board UI** | Kanban columns for each status. • Drag card → status change. • Color ribbon by **division** (DIV A, MED, STAGING). • Sub-badge shows #persons / #resources attached. |
| **Card click** opens **Task Drawer** | Tabs: Details, Checklist (if linked), Map, Chat, Log. Quick buttons: “Arrived”, “Need Assist”, “Complete”. |
| **Link to 204 PDF** | divisionCode + task.id appear in in-app 204 view. Latest **task title/geo** NOT frozen until period closes. |
| **Offline merge** | Status history stored in TaskStatusHistory; merge picks furthest-along state & latest timestamp for non-state edits. |

## 3.2 Resource & Roster linkage

| **Table** | **Purpose** |
| --- | --- |
| **TaskPersonnel** (taskId, personId, role, checkIn, checkOut) |  |
| **TaskResource** (taskId, resourceId, qty) |  |

Adding or removing people/items auto-updates the 204 roster view and triggers notifications.

## 3.3 Field Reporting Loop

| **Action (mobile)** | | **DB change** | | **Auto notification** |
| --- | --- | --- | --- | --- |
| **Arrive on scene** (tap) | | TaskStatusHistory: In-Progress + GPS Task.checkInTime | | Ops & IC get toast if > 15 min late |
| **Quick note / photo** | | TaskComment row + FileBlob if photo | | Followers get push |
| **Need Assistance** | | Status → Blocked + reason code | | Red alert to Ops |
| **Complete task** | | Status → Complete; auto‐stamp completedAt, totalMinutes | | Dashboard metric updates |
| **Demob** | | Ops marks → ‘Demobbed’; roster entries auto check-out | |  |
| **Change status** | Only to one of the *allowed toCodes* from TaskTransition for that task’s typeCode. | | Insert TaskStatusHistory row. | |

|  |  |  |
| --- | --- | --- |
| **Photo attach** | Any state. | File validated image/jpeg or image/png. |

|  |  |  |
| --- | --- | --- |
| **GPS beacon** | Mobile detects task in an **isActive** state → posts /tasks/{id}/beacon every 60 s (± jitter). |  |

Each action works offline; queued events replay with the original timestamp when connectivity returns.

## 3.4 Live Status Dashboards

| **Widget** | **Calculation** |
| --- | --- |
| **Task funnel** | Count by status, color coded. |
| **Overdue tasks** | Now – plannedEnd > threshold AND status ≠ Complete. |
| **Resource utilization** | # personnel assigned / total available. |
| **Objective completion** | For each Objective: completeTasks / totalLinkedTasks. |
| **Heat map toggle** | Shows In-Progress task footprints; opacity = age of last update. |

Dashboards refresh from local cache every 15 s; edge/cloud sync streams deltas via WebSocket when online.

## 3.5 Data model (simplified)

sql

CopyEdit

Task (

id UUID PK,

periodId UUID FK,

divisionCode TEXT,

title TEXT,

description TEXT,

objectiveId UUID FK NULL,

geoType TEXT,

geoData JSONB,

priority CHAR(1), -- A–D

status TEXT, -- enum

checkInTime TIMESTAMPTZ,

completedAt TIMESTAMPTZ,

createdBy UUID,

createdAt TIMESTAMPTZ,

updatedAt TIMESTAMPTZ

);

TaskStatusHistory (

id UUID PK,

taskId UUID FK,

newStatus TEXT,

changedBy UUID,

lat DOUBLE PRECISION NULL,

lon DOUBLE PRECISION NULL,

note TEXT NULL,

stamp TIMESTAMPTZ

);

TaskPersonnel (taskId, personId, role, checkIn, checkOut);

TaskResource (taskId, resourceId, qty);

TaskComment (id, taskId, authorId, text, stamp);

enum\_task\_status: Planned, Briefed, In-Progress, Blocked, Complete, Demobbed.

## 3.6 Sync & conflict rules

| **Field** | **Merge strategy** |
| --- | --- |
| **status** | Use **FSM precedence**: the state furthest right in the flow wins if timestamps are within 2 min; else latest timestamp wins. |
| **description / geoData / priority** | Last-writer-wins on timestamp. |
| **Roster entries** | Set operations (add/remove) merged by personId/resourceId. |
| **Comments & history** | Append-only, no conflicts. |

## 3.7 API sketch

bash

CopyEdit

GET /periods/{pid}/tasks?division=DIV A

POST /tasks (create)

PATCH /tasks/{id} (edit fields)

POST /tasks/{id}/status (body: { newStatus, note })

POST /tasks/{id}/comment

POST /tasks/{id}/attachments

POST /tasks/{id}/personnel (bulk add/remove)

POST /tasks/{id}/resources

GET /tasks/{id}/history

GET /dashboard/ops?incident={iid}

All endpoints accept If-Offline-Queue header; SDK stores and retries.

## 3.8 Notifications

| **Trigger** | **Recipients** | **Medium** |
| --- | --- | --- |
| Task status ⇒ Blocked | Ops + IC | Push + e-mail (optional) |
| Task Complete | Linked Objective followers | Push |
| Overdue threshold crossed | Ops | Push |
| Personnel added to task | That person | Push |

Admins can tune thresholds per incident (Settings → Ops Alerts).

## 3.9 Permissions

| **Role claim** | **Rights on tasks** |
| --- | --- |
| **OpsChief** | Create / edit all, status override, roster edits |
| **Division/Group Supervisor** | Edit tasks within their divisionCode |
| **Responder assigned to task** | Update status (but not division), add comments/photos |
| **Viewer** | Read-only |

Role is derived each time sync occurs by matching the latest 203 org rows.

## 3.10 Integration touch-points

| **Module** | **Data exchange** |
| --- | --- |
| **Module 2 Objectives** | Task links via objectiveId; dashboard shows roll-up. |
| **Module 4 Roster / Resources** | TaskPersonnel pulls from the live roster; demob updates availability. |
| **Module 5 Comms** | Each status change auto-logs a line in CommsLog (214 feed). |
| **Module 6 Mapping** | geoData renders as layer; tap map → opens Task Drawer. |
| **Module 7 Export** | When period closes, latest task roster & status feed into 204 PDFs and 214 Unit Log. |

| **Sub-module** | **Purpose** | **Key entities & fields** | **UI / Workflow highlights** |
| --- | --- | --- | --- |
| **4.1 Personnel Roster & Qualifications** | Maintain an always-current list of people, their certs, availability, and call signs. | Person (id, first/last, agency, phone, email, callSign, status *[available / unavailable / on-scene]*). Qualification (code, name, expiresOn). PersonQual (personId, qualCode, issued, expires). | • Grid & T-card rack with multi-filter.  • Status machine available → en route → on-scene → demob → unavailable.  • CSV import/export. |
| **4.2 Equipment & Supply Tracker** | |  | | --- | |  |  |  | | --- | | **Resource Library** *(bar-coded, offline)* | | **Resource** (id, typeCode, subType, description, assetTag, status, lat/lon)  Type-specific child tables:  • **ResourceVehicle** (VIN, plate, make/model, year, seating, radioCallSign, remarks)  • **ResourceAircraft** (tailNumber, make/model, year, extLoadLbs, repeaterCapable, callSign, baseAirport, owner, contactPhone, remarks)  • **ResourceRadio** (serialNumber, make/model, radioKind PORT/MOB/RPTR/BASE, band, outputWatts, radioAlias, radioId, remarks)  • **ResourceMedical / Tool / UAV / Other** (simple descriptor & remarks) | • Pre-seeded **ResourceType** codes: **VEH, AIR, RAD, MED, TOOL, UAV, OTH**.  • VEH sub-types now include VAN12, VAN15, MINIVAN, TRAILER, etc.  • Bar-code label generator (Code 128).  • Scanner auto-opens resource drawer; “Create new” if unknown. |
| **4.3 Task Assignment Bridge** | Attach roster personnel & equipment to tasks (Module 3). | Bridge tables already in Module 3: TaskPersonnel, TaskResource. | Drag-drop people or gear onto a Task card. Check-in/out times write to the bridge rows; availability auto-updates. |
| **4.4 Facility & Ground Support** | Track ICPs, base camps, staging, helispots, fuel points. | Facility (id, type *[ICP / Staging / Helispot / Fuel / Medical]*, name, lat/lon, capacity, contact). | *Map overlay layer* with facility icons. *QR sheet* prints location cards for signs in the field. |
| **4.5 Demobilization & Return-to-Service** | Make sure resources leave the incident cleanly. | DemobChecklist (id, assetKind RESOURCE/PERSON/TEAM, assetId, formVersionId, completed, signedBy) | • Checklist form linked by ResourceType (e.g., DEMOB-VEH-v1, DEMOB-AIR-v1, DEMOB-RAD-v1).  • Typed-name + PIN signature.  • Completing flips asset status back to **ready** (or **unavailable** if needs repair). |
| **4.6 ICS-Form Outputs** | Auto-fill 203 (select staff), 204 roster sections, 205 frequencies (radio cache), 206 Medical Plan (medics list), 219 T-cards, 215/215A (facility & ground), 232 T-Card manifest. | Uses Form Engine (Module 2) with live roster merge on snapshot. | |  | | --- | |  |  |  | | --- | | • Auto-fills 203, 204 rosters, 205 radios (via 217 link), 206 Med Plan, 219 T-cards, 215/215A, 232 manifest.  • Print-ready PDFs produced at IAP snapshot; bar-code-ready 219. | |

#### 4.1 Personnel specifics

Status machine  
available → en route → on-scene → demob → unavailable → available  
‣ Only **Logs** or **RosterMgr** roles may override the machine.  
‣ Auto-promote to **on-scene** when the user’s device is inside the incident geo-fence and GPS beacons have flowed for > 5 min.

Qualification hierarchy  
Supports parent/child codes so **“Swift-water Tech”** automatically satisfies **“Swift-water Awareness.”**

Expiry alerts  
A daily cron job writes an “Expiring in 30 days” alert into the Notification queue; no real-time pop-ups to volunteers.

#### 4.2 Equipment specifics

| **Resource type examples** | **Sub-fields** |
| --- | --- |
| Vehicle | VIN, license, seating, fuel type, 4×4 (bool), mileage. |
| Radio | Band, model, serial, encryption keyset id. |
| Medical kit | Contents list, exp dates, weight. |
| Drone/UAV | FAA reg #, battery count, controller id. |

Status machine mirrors Personnel: ready → en route → deployed → demob → oos → ready.

Maintenance auto-generates tasks when **usage-based** triggers (hours/miles) or **time-based** triggers (annual inspection) are hit.

#### 4.3 Integration with Module 3

* Dragging a **RESCUE 42 Tripod** onto Task #123 writes a TaskResource row (qty=1) and flips its status=deployed.
* When Task #123 hits **Demobbed**, a wizard asks “Return all attached resources to Ready?” (default Yes).
* Task drawer shows **people & gear chips**; long-press pops a quick check-out menu.

## 4.3 Demob checklists per type

| **Resource type** | **Checklist form ID** | **Required signature** |
| --- | --- | --- |
| **Vehicle** | DEMOB-VEH-v1 | Demob Officer + Driver |
| **Radio** | DEMOB-RAD-v1 | Comms Unit Leader |
| **Medical** | DEMOB-MED-v1 | Medical Unit Leader |
| **Drone** | DEMOB-UAV-v1 | Air Ops Supervisor |
| **Tool** / Other | DEMOB-GEN-v1 | Logs Chief |

Engine behavior

1. Logs clicks “Start Demob” on a resource.
2. App looks up resource.typeCode → loads the matching checklist schema.
3. Checklist shows bar-code at top for positive ID; critical items must be ticked before “Release” enables.
4. Signature widget uses **typed name + PIN** (same as Module 2).
5. Completing the form auto-flips resource.status to **ready** (or **unavailable** if the checklist outcome is “needs repair”).
6. Form is exported into the period’s Demob Packet PDF bundle.

## 4.4 Updated ER snippets

sql

CopyEdit

-- Resource categories

CREATE TABLE ResourceType (

code TEXT PRIMARY KEY, -- 'VEH', 'RAD', 'MED', 'TOOL', 'UAV', 'OTH'

name TEXT NOT NULL

);

-- Resource main

ALTER TABLE Resource

ADD COLUMN typeCode TEXT REFERENCES ResourceType(code),

ADD COLUMN assetTag TEXT UNIQUE,

ADD COLUMN barcodeSymbology TEXT DEFAULT 'CODE128';

-- Demob linkage

CREATE TABLE DemobChecklist (

id UUID PRIMARY KEY,

resourceId UUID REFERENCES Resource(id),

formVersionId UUID REFERENCES FormVersion(id), -- from Module 2

completed BOOLEAN DEFAULT false,

completedAt TIMESTAMPTZ,

signedBy UUID

);

Resource-type seeding:

sql

CopyEdit

INSERT INTO ResourceType (code,name) VALUES

('VEH','Vehicle'),('RAD','Radio'),('MED','Medical'),

('TOOL','Tool / Equipment'),('UAV','Drone / UAV'),('OTH','Other');

#### 4.4 Facility & Ground Support

Helispot form (custom) captures LZ coords, elevation, approach azimuth, load limits, contact.  
Fuel depot tracks current diesel/gas qty; analytics tab shows **burn rate**.

#### 4.5 Demob workflow

1. Logs marks a resource/person Demob Ready in the roster.
2. Demob Officer signs digital checklist (Module 7 custom form).
3. Resource moves to **Demob Plan** board (similar to Task board).
4. When released, demobCheckOut is stamped and **resource.status = ready** (or unavailable if heading home).

#### 4.6 Offline & sync considerations

* Roster, resource, facility, and maintenance tables live in device SQLite and merge with set-style semantics (no merge conflicts on numeric counters, last-writer on text).
* Large photo attachments for equipment condition are stored in the same object store as Task photos; queued upload if offline.
* Beacon cadence rules from Module 3 apply **only** to Tasks, not general personnel GPS.

Demobilization

## 1 Extended Demob Checklist model

sql

CopyEdit

CREATE TYPE DemobAssetKind AS ENUM ('RESOURCE','PERSON','TEAM');

CREATE TABLE DemobChecklist (

id UUID PRIMARY KEY,

assetKind DemobAssetKind NOT NULL,

assetId UUID NOT NULL, -- FK to Resource.id, Person.id, or Team.id

formVersionId UUID REFERENCES FormVersion(id),

completed BOOLEAN DEFAULT false,

completedAt TIMESTAMPTZ,

signedBy UUID

);

CREATE INDEX ON DemobChecklist (assetKind, assetId);

### Template mapping

| **Asset kind** | **Template code (example)** | **Signature role** |
| --- | --- | --- |
| RESOURCE:AIR | **DEMOB-AIR-v1** | Air Ops Sup. |
| RESOURCE:VEH | DEMOB-VEH-v1 | Demob Off. + Driver |
| PERSON | **DEMOB-PER-v1** | Section Chief |
| TEAM | **DEMOB-TEAM-v1** | Division/Group Sup. |

Admin UI lets DB Admins bind any checklist form to each asset kind (and, for resources, to each ResourceType).

## 2 Workflow changes

### Personnel

1. Logs (or Section Chief) presses **“Start Demob”** on a person in the roster.
2. App opens DEMOB-PER-v1. Critical items (gear return, medical out-brief, etc.) must be ticked.
3. Signature (typed name + PIN) → Person.status ↦ unavailable (or ready if agency SOP).

### Teams

Teams are represented by the **Org Chart “branch root” rows** (e.g., DIV A, Strike Team 1).  
Logs can right-click the branch → **Demob Team** → DEMOB-TEAM-v1 launches.  
Completing it bulk-demobs all attached personnel & equipment (with override option).

### Aircraft resources

Resource.typeCode='AIR' triggers **DEMOB-AIR-v1** when Logs demobs that tail #.  
Status flow identical to VEH (ready → en route → deployed → demob → ready).

## 3 UI tweaks

* **Roster Grid** gains a **Demob** column (checkbox) for quick multi-select → bulk demob.
* **Team Demob Board** (new) shows progress bars for every active team.
* **Resource Drawer** for aircraft shows extra fields (tail #, hours, fuel).

## 4 Admin tooling

* **Demob Form Binder** page lists each assetKind (and each ResourceType) with its currently linked checklist form version; DB Admins can change linkage at any time.
* Aircraft resource-type editor exposes extra metadata fields and maintenance triggers (flight hours).

## 5 Next actions I’ll incorporate into the scaffold

1. **DB migration** adding DemobAssetKind, DemobChecklist, and seeding ResourceType 'AIR'.
2. **Custom-form stubs** for DEMOB-PER-v1, DEMOB-TEAM-v1, and DEMOB-AIR-v1 (critical/optional items, signature block).
3. **Demob wizard** logic extended to detect asset kind and launch the right form.
4. **Admin binder UI** for mapping forms to asset kinds / resource types.

**Aircraft record — add “Repeater-Capable” flag**

| **Field (new)** | **Type** | **Example** | **Purpose** |
| --- | --- | --- | --- |
| repeaterCapable | **boolean** | true | Quickly shows whether the aircraft can carry or has an installed airborne radio repeater for comms relay. Appears as a green ✓ / red ✕ icon in lists and on the map. |

 **Aircraft drawer**: shows a toggle switch **“Repeater-capable”**; only users with Roster / Resource Edit rights can change it.

 **Resource grid filter**: new quick filter **Repeater = Yes** to help Comms Unit locate suitable platforms.

 **Map overlay**: aircraft icons with repeater-capable flag get a small stacked-antenna badge.

 **ICS-205A export** (optional future): if an aircraft with repeaterCapable = true is assigned to a comms task, the flag can auto-populate the remarks column.

Module 5 — **Communications & Logs**

(full-stack design draft, ready for you to trim, rename, or extend)

| **Sub-module** | **Purpose** | **Primary tables & key fields** | **UI / Workflow highlights** |
| --- | --- | --- | --- |
| **5.1 Radio / Message Log** | Capture every significant radio, phone, or in-person message so it can roll into ICS-214 Unit Log and after-action reports. | CommsLog • id UUID • periodId FK • ts timestamp (auto-UTC) • channel (short text, e.g. “Ops 3”) • frequency (text 136.650 MHz or TG ID) • fromUnit (text or personId FK) • toUnit (text or personId FK) • message text (≤ 280 chars quick-log, unlimited in full edit) • attachmentRef FK → FileBlob (optional photo / doc) • enteredBy userId | Quick-Entry Panel on mobile / desktop: 1️⃣ Tap a recent channel (chips flex with last 6 used). 2️⃣ Mic pops to capture short voice → converts to text (optional future). 3️⃣ Tap ✔️ to save; entry appears instantly in the period-wide log. Search & filter grid (channel, from, string). Export: last save wins, then auto-pulled into ICS-214 when the period closes. |
| **5.2 Threaded Chat** | Provide Slack-style chat but scoped to Incident, Period, or Task, and remain offline-tolerant. | ChatChannel (scope incident/period/task)  ChatMessage (id, channelId, authorId, ts, text) | Slack-style channels (incident-wide, per period, per task). Offline queue; last 200 messages cached per channel. @mentions push a device notification. |
| **5.3 Comms Plan Builder** | Build & version the official **ICS-205 / 205A** radio & phone plan for the current period. | CommsPlan • periodId, version, authorId, createdAt, payloadJSON, signature CommsChannel (embedded in payload or separate table) • lineNo, function, channelName, rxFreq, txFreq, rxTone, txTone, band, notes | Wizard layout copies the FEMA PDF grid: Add channel autosuggests from ResourceRadio.band + channel library. Validation: dup freq warnings, tone field required if CTCSS present. Signing uses typed-name + PIN (Module 2 widget). Publish Snapshot: latest version goes into the IAP PDF; editing still allowed until the period closes (per your lock-after-close rule). |
| **5.4 Integration hooks** | Keep everything in sync with other modules. | — | • New CommsPlan versions push alerts to **CommsUnit** role. • Task status changes auto-log to CommsLog. • Live “Comms Feed” dashboard card shows last 10 log lines & msgs/min rate. • All exports (214/217/205/309) appear in **Forms** tab and archive ZIP at incident close. |

## 5.1 Radio / Message Log — design notes

* **Timestamp** is device-local but converted to UTC at save, so offline entries keep true incident time.
* **Attachments** are optional (photo of scratch pad, screenshot of phone SMS). **Voice recordings are not enabled by default**—keeps storage simple and avoids policy issues.
* **Quick-Entry hotkeys** on desktop: Ctrl+Shift+N pops the log dialog anywhere in the app.

|  |  |
| --- | --- |
| **Only “Channel Name” is required** | CommsLog now keeps **channel** (text) and **message**. Columns frequency, fromUnit, toUnit are dropped (they can be re-added later if you want them). |
| **Voice capture & file attachments deferred** | • Columns for attachmentRef and any WAV/MP3 upload endpoints are removed from MVP and added to the **Future Features Backlog**. |
| **Entry classification for 214 export** | Add entryType ENUM: OPS (Ops traffic) • COMMAND (IC/Section Chief directions) • INFO (general chatter). A boolean view includeIn214 = (entryType IN ('OPS','COMMAND')). Only those rows feed the ICS-214. |
| **Full log → ICS-309 export** | *Export → “Generate ICS-309”* produces the complete period (or date-range) log as PDF/CSV. |

### Table snip

## CREATE TABLE CommsLog (

## id UUID PRIMARY KEY,

## periodId UUID REFERENCES OperationalPeriod(id),

## ts TIMESTAMPTZ NOT NULL,

## channel TEXT NOT NULL,

## fromUnit TEXT, -- ★ new

## toUnit TEXT, -- ★ new

## entryType TEXT NOT NULL CHECK (entryType IN ('OPS','COMMAND','INFO')),

## message TEXT NOT NULL,

## enteredBy UUID REFERENCES "User"(id),

## systemGenerated BOOLEAN DEFAULT false

## );

## 5.2 Threaded Chat — key behaviours

* Messages ≤ 4 000 chars.
* On first boot offline, cached last 200 messages per channel; older fetch when scrolled.
* Period closing **does not lock chat**; it simply rolls messages into the 214 export snapshot (one big appendix).

## 5.3 Comms Plan Builder — versioning flow

| **Step** | **Actor** | **Result** |
| --- | --- | --- |
| Draft plan v1 | Comms Unit Ldr | CommsPlan.version = 1, draft |
| Edit rows | same or delegate | each save bumps minor rev in UI; DB keeps single JSON blob until publish |
| **Publish** | Section Chief signs | v1 marked final, PDF pushed into snapshot |
| Later fix (same period) | Unit Ldr → “Create v2 from v1” | New row version = 2; publish when done |
| Close period | System auto-publishes if last publish > 15 min old and locks both v1/v2. |  |

### 5.3.2 205 Builder now pulls from 217

* **Add Channel** dialog shows a pick-list of **available ICS-217 resources** plus a “Create ad-hoc” option.
* When a row is used in a 205, its 217 **status flips to “ASSIGNED”** (re-usable next period unless you mark OOS).
* 205 PDF carries the chosen channelName + freq/tone data; no need to re-type.

Schema:

CREATE TABLE CommsResource217 (

id UUID PRIMARY KEY,

incidentId UUID NULL REFERENCES Incident(id),

channelName TEXT NOT NULL,

band TEXT,

rxFreq TEXT,

txFreq TEXT,

rxTone TEXT NULL,

txTone TEXT NULL,

nacOrSlot TEXT NULL,

powerWatts SMALLINT NULL,

systemOwner TEXT, -- ★ NEW

status TEXT CHECK (status IN ('AVAILABLE','ASSIGNED','OOS')) DEFAULT 'AVAILABLE',

resourceNotes TEXT

);

payloadJson mirrors the FEMA 205/205A grid so future custom column order is painless.

## 5.4 Dashboards & exports

* **Live Comms Feed panel** for IC/Plans shows last 10 log entries plus message rate/min.
* **Searchable archive** — CommsLog indexed; filtering by date, channel, or keyword in after-action.
* **Exports**
  + Period close ⇒ all CommsLog & ChatMessage lines flatten into the **ICS-214 PDF** (auto-generated by Module 7).
  + CSV/JSON dump available anytime for import to EOC CAD.

| **Document** | **Source rows** |
| --- | --- |
| **ICS-214** | CommsLog.entryType IN ('OPS','COMMAND') + chat messages you flag manually |
| **ICS-309** | *All* CommsLog rows for the selected period/date-range |
| **ICS-205 / 205A** | Latest **CommsPlan** version, with channels pulled from **CommsResource217** |
| **ICS-217** | Direct print/export of the resource table (current status column included) |

**Comms Log → ICS-214: new “select-by-unit *and* message-category” rules**

Below is the redesign that puts **each unit in charge of what flows into its own ICS-214** and uses a clear **message-category flag** (e.g., *Retasking*, *Status Change*, *Needs Assistance*) to decide inclusion.

| **Change** | **Implementation** |
| --- | --- |
| **1 . Message category field** | Add **msgCategory** ENUM to CommsLog with these starter values: • STATUS\_CHANGE • RETASKING • NEEDS\_ASSIST • INFO (everything else) Quick-entry panel shows a small dropdown; system-generated task updates pre-tag themselves (e.g., status changes). |
| **2 . Unit-scoped 214 export** | When someone clicks **Export → ICS-214**, the dialog now asks **“Which unit log?”** and offers: • any *Org-Chart* position (e.g., “DIV A Sup”, “Medic 2”) • custom free-text unit if needed. |
| **3 . Inclusion rule** | For the chosen *unitName* **U** the exporter includes a CommsLog row if (fromUnit = U OR toUnit = U) AND msgCategory IN ('STATUS\_CHANGE','RETASKING','NEEDS\_ASSIST') |
| **4 . Full 309 export unchanged** | **ICS-309** still dumps *all* rows (no filtering). |
| **5 . Schema delta** |  |

sql

CopyEdit

ALTER TABLE CommsLog

ADD COLUMN msgCategory TEXT NOT NULL

CHECK (msgCategory IN ('STATUS\_CHANGE',

'RETASKING',

'NEEDS\_ASSIST',

'INFO'))

DEFAULT 'INFO';

**Automatic “Status-Change” logging—now baked into both the Comms Log and the ICS-214 workflow**

| **What happens** | **How it’s implemented** | **Why it satisfies your rules** |
| --- | --- | --- |
| **A unit’s status changes**  • User taps *“En-Route”* or *“Complete”* in the mobile app **-or-**  • Ops/IC drags the card to a new column on the status board | **Event hook in Module 3** fires:  ts // pseudo<br>onStatusChange(taskOrPerson, oldStatus, newStatus, actorId) {<br> insert into CommsLog {<br> periodId: currentPeriodId,<br> ts: nowUTC(),<br> channel: 'Status',<br> fromUnit: taskOrPerson.displayName, // e.g., "Rescue 31"<br> toUnit: 'Command', // default target; editable later<br> msgCategory: 'STATUS\_CHANGE',<br> message: `${oldStatus} → ${newStatus}`,<br> enteredBy: actorId, // user or 'system'<br> systemGenerated: true<br> };<br>} | *Every* status flip is captured—even offline (queued locally, synced later).  The msgCategory is **STATUS\_CHANGE**, which your 214 export rule already includes. |
| **214 Export** | Export dialog asks **“Which unit log?”**. If you pick *“Rescue 31”*, the exporter selects rows where (fromUnit = 'Rescue 31' OR toUnit = 'Rescue 31') **AND** msgCategory = 'STATUS\_CHANGE' (plus RETASKING / NEEDS\_ASSIST if desired). | Status changes now appear automatically in that unit’s 214 PDF—no manual typing. |
| **Comms Log view** | Grid shows a grey bolt ⚡ icon in the left margin for systemGenerated=true rows so ops can spot auto-generated entries. | Keeps the feed readable and still editable (users with proper role can add context or correct the message text). |
| **Full ICS-309 export** | Continues to dump **all** CommsLog rows—including these status-change lines—so the complete record is preserved. | Meets your requirement for a comprehensive 309. |

**Automatic, per-unit ICS-214 generation**

Your 214 workflow now becomes **completely automatic**—one log per unit, no dialog or manual trigger.

| **Phase** | **What happens behind the scenes** |
| --- | --- |
| **During the period** | *Every* status change (or any other system-generated entry) is written to **CommsLog** with: • fromUnit = <unit name> (e.g. “Rescue 31”) • msgCategory = STATUS\_CHANGE (or RETASKING / NEEDS\_ASSIST) Users can still add human-typed OPS / COMMAND lines. |
| **Publish IAP snapshot**  *(or at period close, whichever comes first)* | The 214 engine: 1️⃣ Scans **CommsLog** for each unique **unit U** that appears in *fromUnit* or *toUnit*. 2️⃣ Collects rows where (fromUnit = U OR toUnit = U) **AND** msgCategory IN ('STATUS\_CHANGE','RETASKING','NEEDS\_ASSIST'). 3️⃣ Fills the **ICS-214 PDF** for that unit, using: • Block 3 “Unit / Division / Group Name” → *U* • Block 8 “Remarks” grid → timestamp + message text. 4️⃣ Saves the PDF at /iap-snapshots/{periodId}/214-{unit}.pdf and records it in FormInstance. |
| **Viewing logs** | The **Forms** tab automatically lists an **ICS-214** entry for every unit that generated at least one qualifying line. Plans / IC can preview or download without extra clicks. |
| **Full 309** | *Generate ICS-309* still dumps **all** CommsLog rows (no filtering) for archive or upload to the EOC. |

### Module 5.3.1 — ****ICS-217 “Communications Resource Availability”****

A self-contained library of every channel or talk-group the incident might use. It feeds the ICS-205 builder and gives the Comms Unit a single, locked-down source of truth.

## 1 What the 217 module does

| **Capability** | **How it works** |
| --- | --- |
| **Master list of channels / talk-groups** | Each row in the grid is one resource (simple simplex channel, repeated pair, trunked TG, phone bridge, etc.). |
| **Reusable library** | Rows can be agency-wide (no incidentId) **or** copied into a specific incident so you can prune and edit without touching the master list. |
| **Status tracking** | AVAILABLE / ASSIGNED / OOS column shows at a glance what’s free, in use, or down for the count. |
| **Validation & duplicate guard** | Inline checks for duplicate channel names or conflicting Rx/Tx frequencies in the same band. |
| **Lock after initial load** | The moment the first row is saved, the grid flips to read-only for everyone except **DBAdmin**. DBAdmin sees an **Unlock** button (audit-logged) to make rare corrections. |
| **Feeds the 205 wizard** | In the ICS-205 builder, the “Add Channel” dialog lists **AVAILABLE** rows first. Choosing a row immediately flips its status to **ASSIGNED**. |
| **Exports** | • “Print ICS-217” button renders the grid into FEMA PDF layout (filterable by status). • CSV export for programming software. |

## 2 Current schema (with your latest tweaks)

sql

CopyEdit

CREATE TABLE CommsResource217 (

id UUID PRIMARY KEY,

incidentId UUID NULL REFERENCES Incident(id), -- NULL = global library row

channelName TEXT NOT NULL, -- "V-TAC11"

band TEXT, -- "VHF", "UHF", "700/800", "HF"

rxFreq TEXT, -- "155.7875"

txFreq TEXT, -- may be same for simplex

rxTone TEXT NULL, -- CTCSS/DPL

txTone TEXT NULL,

nacOrSlot TEXT NULL, -- P25 NAC, DMR slot, etc.

powerWatts SMALLINT NULL, -- 0-350

systemOwner TEXT, -- "County SAR", "NIFC", etc. ★

status TEXT CHECK (status IN

('AVAILABLE','ASSIGNED','OOS')) DEFAULT 'AVAILABLE',

resourceNotes TEXT

);

Indexes: (incidentId, status), (band, rxFreq, txFreq) for fast look-ups and duplicate detection.

## 3 UI & workflow

### 3.1 Grid view (desktop & tablet)

| **Column** | **Behaviour** |
| --- | --- |
| **Ch # / Name** | Primary key; editing duplicates triggers red highlight. |
| **Band** | Dropdown with VHF, UHF, 700/800, HF, SAT. |
| **Rx / Tx Freq** | Inline numeric field; auto-formats to 4 decimal places. |
| **Rx / Tx Tone / NAC** | Tone fields appear / hide based on band. |
| **Power (W)** | Optional — grey placeholder when blank. |
| **System / Owner** | Free-text; shown in 205 pick-list. |
| **Status** | Badge (green ✓ Available, blue ⇄ Assigned, red ✕ OOS). |
| **Notes** | Hover preview; click to edit. |

### 3.2 Editing

* **Add Resource** button opens a modal; on **Save** the entire grid locks (unless DBAdmin).
* **Bulk import** accepts CSV → shows preview → “Import & Lock”.

### 3.3 Integrating with ICS-205 builder

1. Builder opens → hits /comms/217?incidentId=X&status=AVAILABLE.
2. Pick-list shows rows with **Channel Name, Band, System/Owner**.
3. Selecting a row clones its freqs/tones into the 205 grid and flips the source row’s status to **ASSIGNED** (so nobody double-books a channel).
4. Deleting a row from the 205 (or cloning the plan to the next period) toggles the 217 status back to **AVAILABLE** automatically.

### 3.4 Lock / unlock rules

| **Role** | **Default rights** |
| --- | --- |
| **CommsUnit** | Can view, filter, export. |
| **Plans** | View only. |
| **DBAdmin** | Unlock for edit (audit-logged). |

Unlock operation writes to AdminAuditLog with user, timestamp, and reason string.

## 4 Exports

* **ICS-217 PDF** — Matches FEMA boxes:<br> • Block 1 Incident Name auto-fills.<br> • Block 2 “Date” = snapshot time.<br> • Channel grid prints only rows with status != OOS unless you tick “Include OOS”.
* **CSV** — Column order = channelName,band,rxFreq,txFreq,…; 205 builder can import another agency’s CSV directly.

## 5 Future features backlog (not in MVP)

1. **Voice recorder & file attachments** — once you green-light attachments we’ll allow docs (channel license PDFs, coverage maps).
2. **Geo-coverage polygons** per channel to display shaded map overlays.
3. **Inter-incident sharing** — push a read-only 217 view to cooperating agencies via secure link.

### Adding ****Line A**** restrictions (lower-48/Canada) alongside the existing ****Line C**** rules

The mechanism we just built—an authorizedZone polygon (or point + radius) on each **ICS-217** row—works for **both** regulatory boundaries:

| **Boundary** | **In FCC Part 90** | **Where it applies** | **What NIFOG says** |
| --- | --- | --- | --- |
| **Line A** | 47 CFR §90.7 | A curved line roughly **75 mi (120 km) south of the 49th-parallel** Canada/US border, stretching from WA to ME. Any VHF/UHF land-mobile transmitter **north of the line** (i.e., closer to Canada) needs Industry Canada coordination. | All NIFOG VTAC/UTAC/7-TAC channels foot-noted “LINE A” are prohibited **north of Line A** unless you have formal coordination. |
| **Line C** | 47 CFR §90.7 | Alaska / Yukon: transmitters **within 200 mi (320 km) of Line C** (a boundary inside AK) need coordination. | VTAC/UTAC channels marked “C-LINE” have the same rule in Alaska. |

## 1 How we’ll seed the polygons

| **File** | **Contents** | **Loaded into DB as…** |
| --- | --- | --- |
| line\_a\_restriction.geojson | A single **MultiPolygon** covering every U.S. county **north of Line A** (including the 75 mi buffer). | Row id='BOUNDARY-LINEA' in a helper table RegulatoryZone. |
| line\_c\_restriction.geojson | Polygon covering the 200-mile coordination band inside Alaska. | Row id='BOUNDARY-LINEC'. |

Both files use WGS-84 EPSG 4326 coordinates and can be edited later if the FCC definitions change.

## 2 Linking a restricted channel row

When DB Admin (or Comms Unit) creates / imports a channel that carries the **LINE A** foot-note:

1. In the **217 editor** choose **Set geographic restriction → “Line A (FCC region)”**  
   – UI simply copies the authorizedZone geometry from RegulatoryZone('BOUNDARY-LINEA').
2. **restrictionNote** is auto-filled →  
   “LINE A – Canada coordination required for use north of the line.”
3. Save → row becomes locked like any other 217 entry.

Same flow for a “C-LINE” channel with BOUNDARY-LINEC.

## 3 Validation logic (unchanged, now with two built-in zones)

sql

CopyEdit

-- Pseudocode in builder service

IF channel.authorizedZone IS NOT NULL THEN

allowed := ST\_Contains(channel.authorizedZone, icpPoint);

IF NOT allowed THEN

raise "Channel restricted by "||channel.restrictionNote;

END IF;

END IF;

The ICP’s **lat/lon** is the point; authorizedZone already references the correct polygon.

## 4 Edge-cases & overrides

| **Scenario** | **Behaviour** |
| --- | --- |
| ICP moves during incident | Builder re-validates all ASSIGNED channels; restricted rows throw an alert so the Comms Unit can swap them out. |
| Plans Chief truly has Canadian coordination | Plans Chief (role) may click **Override** in the 205 grid. Override is PIN-protected, audit-logged, and places a **★ foot-note** on the 205 PDF (“Coordination #2025-C-1234 on file”). |
| Mobile/task radios operating temporarily across the border | Still governed by where the **transmitter** is. If you dispatch a repeater north of Line A, the system will catch it when you set its GPS or manually assign its location. |

## 1. Per-channel data fields (grid columns)

| **Field** | **What you enter / choose** | **Notes** |
| --- | --- | --- |
| **Channel Name or ID** | Free text (e.g., “V-TAC 11”) | Duplicate-check highlights conflicts. |
| **Band** | VHF 136-174 MHz • UHF 400-520 • 700/800 • HF • SAT | List can be extended in lookup table. |
| **Rx Freq** | Numeric (4 decimals) | For simplex just repeat in Tx box. |
| **Tx Freq** | Numeric | — |
| **Rx Tone** | CTCSS / DPL (optional) | Hidden for HF unless you toggle. |
| **Tx Tone** | Same | — |
| **NAC / Slot** | P25 NAC (e.g. “$293”) or DMR slot | Optional. |
| **Power (W)** | 1 – 350 (optional) | Helps the 205 “ERP” note. |
| **System / Owner ID** | e.g. “County SAR”, “NIFC”, “USFS-R5” | Added per your request. |
| **Status** | **AVAILABLE** (green) • **ASSIGNED** (blue) • **OOS** (red) | Auto-flips to **ASSIGNED** when you pull it into the 205; flips back when released. |
| **Restriction Note** | Free text shown on hover | “LINE A – Canada coordination required”. |
| **Geographic Restriction** | ­🔒 Polygon **or** Point + Radius | Pre-sets for **Line A** and **Line C**; you can draw custom zones. |
| **Resource Notes** | Long text | Misc. remarks (licence, antenna site, etc.). |

## 2. Library scope options

* **Agency-wide row** (incidentId = NULL) – lives forever, visible in every new incident.
* **Incident copy** – clone any subset of rows into an incident-specific 217 so you can prune or mark OOS without touching the master list.

## 3. Editing & locking controls

| **Role** | **Can do** |
| --- | --- |
| **CommsUnit / Plans** | Add rows **until first Save**; after that the grid is read-only. |
| **DBAdmin** | “Unlock for edit” (audit-logged) ✅. |
| Everyone | Filter, sort, export PDF/CSV. |

## 4. Validation & duplicate protection

* **Duplicate channel name** or identical Rx/Tx pair in same band → red highlight, save blocked.
* **Geo check** – When a channel has a restriction polygon:
  + If the ICP point is **outside** zone → save allowed.
  + **Inside** zone → error banner: “Channel restricted by Line A.”
  + Plans Chief can PIN-override in the 205 (audit-logged, PDF footnote).

## 5. Integration touch-points

1. **ICS-205/205A builder**
   * Add channel pick-list sorts **AVAILABLE** rows first.
   * Selecting a row sets its status → **ASSIGNED** automatically.
   * Deleting or cloning a 205 row flips it back to **AVAILABLE**.
2. **Snapshot / IAP export**
   * 217 rows print into a FEMA-layout PDF (“Print ICS-217”)—optionally filter out OOS rows.
3. **Audit log**
   * Unlocks, overrides, and bulk imports are all time-stamped with user & reason.

## 6. Future-feature placeholders (disabled for MVP)

* **File attachments** (licence docs, coverage maps).
* **Coverage polygons** shown on the incident map.
* **Inter-incident sharing link** (read-only 217 for partner agencies).
* **Automated voice-record capture** items in the Comms Log.
* **ICS-217 updates based on your latest instructions**

| **Change** | **How we’ll implement it** |
| --- | --- |
| **1. Mode A/D/M field** | New column **modeADM** ('A' = Analog · 'D' = Digital · 'M' = Mixed) added to the grid—single-letter dropdown. |
| **2. Geographic restriction limited to Line A / Line C only** | • Custom-draw polygons are removed.  • Column **geoRestriction** replaces authorizedZone and is an **ENUM**: 'NONE' |
| **3. Remove Power column** | Field **powerWatts** is deleted from the schema and UI. |
| **4. Band list trimmed** | Dropdown now offers **VHF · UHF · 700/800 · HF** (SAT removed). |

### Revised table definition

sql

CopyEdit

CREATE TABLE CommsResource217 (

id UUID PRIMARY KEY,

incidentId UUID NULL REFERENCES Incident(id),

channelName TEXT NOT NULL,

band TEXT CHECK (band IN ('VHF','UHF','700/800','HF')),

modeADM TEXT CHECK (modeADM IN ('A','D','M')), -- ★ new

rxFreq TEXT,

txFreq TEXT,

rxTone TEXT NULL,

txTone TEXT NULL,

nacOrSlot TEXT NULL,

systemOwner TEXT,

geoRestriction TEXT -- ★ new ENUM-like field

CHECK (geoRestriction IN ('NONE','LINE\_A','LINE\_C'))

DEFAULT 'NONE',

status TEXT CHECK (status IN ('AVAILABLE','ASSIGNED','OOS')) DEFAULT 'AVAILABLE',

resourceNotes TEXT

);

-- (powerWatts column removed)

Existing deployments: run ALTER TABLE commands to drop powerWatts, add modeADM (default 'A'), and add geoRestriction (default 'NONE').

### Validation logic

* **Line A / Line C enforcement** now checks geoRestriction value:  
  If *geoRestriction = 'LINE\_A'* and ICP is north of Line A → block / require override.  
  Same for 'LINE\_C' with the Alaska boundary.
* No other polygons can be created.

## 1  Schema additions

sql

CopyEdit

-- Replace single-value column with an ARRAY of text values

ALTER TABLE CommsResource217

DROP COLUMN eligibleUsers,

ADD COLUMN eligibleUsers TEXT[] NOT NULL

DEFAULT ARRAY['ANY']::TEXT[];

-- Constraint: every array element must be in the allowed set,

-- and 'ANY' may not coexist with another value.

ALTER TABLE CommsResource217

ADD CONSTRAINT chk\_217\_eligible\_users

CHECK (

eligibleUsers <@ ARRAY[

'ANY','PUBLIC\_SAFETY','AMATEUR','SAR\_VOLUNTEER','COMMAND\_ONLY'

]::TEXT[]

AND NOT ('ANY' = ANY(eligibleUsers) AND cardinality(eligibleUsers) > 1)

);

 Values are stored as a **text array** (TEXT[]).

 Allowed set: ANY, PUBLIC\_SAFETY, AMATEUR, SAR\_VOLUNTEER, COMMAND\_ONLY.

 If ANY is present it must be the **only** value.

You can extend the list in a future migration; code looks up allowed values from a lookup table.

## 2  Editor UI

| **Field** | **Widget** |
| --- | --- |
| **Eligible Users** | Dropdown (**Any • Public-Safety • Amateur • SAR Volunteer • Command-Only**) |

Rows default to **Any**. Selecting Amateur automatically appends “🛠️ Ham-only” badge in the grid.

## 3  Behaviour in other modules

| **Where** | **What happens** |
| --- | --- |
| **ICS-205 builder pick-list** | Channel rows show the eligible-users badge; if you pick Amateur the row foot-note in the 205 reads “Licensed ham operators only.” |
| **Status / Task board auto-assignment hooks** | No hard block for now, but if you later link **PersonQual** (“HAM\_LICENSE”) we can warn when a non-eligible person is assigned. |
| **PDF exports** | • ICS-217: extra column “Eligible Users”. • ICS-205: star-footnote on channels with anything other than **Any**. |
| **Search / filter** | New filter pill in the 217 grid and the 205 pick-list: Any • Public-Safety • Amateur • … |

## 4  Lock-and-validation rules

* Field is editable until the 217 grid first locks; after that only **DBAdmin → Unlock** can change it.
* Validation: if **eligibleUsers='AMATEUR'** and **band='700/800'** (not allowed under Part-97) the editor flashes a red error.

Module 6 — **Mapping & Situational Awareness**

(current design snapshot – ready for refinement whenever you are)

| **#** | **Sub-module** | **Core entities & fields** | **Highlights / Workflow** |
| --- | --- | --- | --- |
| **6.1** | **GIS Engine & Tile Cache** | • MapTile cache dir (MBTiles / ZIP)  • LayerCatalog (id, name, type XYZ/WMS/GeoJSON, sourceURL, isOffline) | Offline-first: every device keeps a local MBTiles bundle (USGS Topo, NAIP Aerial). Layer toggle drawer shows Base layer + Overlay check-boxes (perimeter, ICP icons, facilities, 217 polygons). |
| **6.2** | **Live Tracking (Beacon ingest)** | TrackPoint (id, senderId, senderKind TASK/PERSON/RESOURCE, tsUTC, lat, lon, alt) | Mobile SDK posts a **TrackPoint** every N seconds where N = beacon interval set per **TaskType** (Module 3). Map displays: • Last-known position icon with status color. • Breadcrumb tail (fade after 12 h). |
| **6.3** | **Search-Planning Toolkit** | SearchSegment (id, incidentId, name, geometry Polygon/Line, probArea, probDetect) | • Draw / auto-grid tool (size by acres or distance). • Assign segment to tasks (link to taskId). • Coverage heat-map colors by probability of detection × probability of area. |
| **6.4** | **Geofences & Alerts** | GeoFence (id, incidentId, name, geometry, alertWhen IN/OUT, linkedTaskId?) | • Example: “No-Fly Zone” around helibase triggers toast if drone/UAV beacon enters. • Fence status updates appear in Comms Log (msgCategory=INFO). |
| **6.5** | **Facility & Ground-Support Overlay** | Re-uses Facility rows from Module 4 | ICP, Staging, Helispots, Fuel, Medical icons auto-layered; click pops resource drawer. |
| **6.6** | **Snapshot & Export** | Map snapshot saved with every IAP (PNG + worldfile) | IAP PDF shows situational-awareness inset page with perimeter + task icons. |
| **6.7** | **Analytics Dashboards** | Live cards: “Active Beacons / min”, “Last update per team”, “Segments covered %” | Pulls from TrackPoint & SearchSegment tables. |

#### Key design notes

* **Coordinate system** — all geometry stored in WGS-84 (EPSG 4326); heavy spatial queries use PostGIS.
* **Tile management** — admin CLI tiles pull usgs-topo 45,-90 49,-84 to cache new AOI; devices auto-sync tiles when on Wi-Fi.
* **Bandwidth guard** — TrackPoints are queued and chunk-uploaded when offline; max 1 KB/min per active device at default cadence.
* **Beacon styling**
  + Task beacons inherit task-status color (Planned = grey, In-Progress = blue, Needs-Assist = red).
  + Personnel beacons use role icon (medic, K-9, etc.).

#### Integration touch-points

| **With module** | **Data flow** |
| --- | --- |
| **Module 3 Task board** | Task start ⇒ Map auto-centers on assigned segment; status change inserts TrackPoint if senderKind = TASK. |
| **Module 4 Resources** | Aircraft and vehicles with GPS feed post TrackPoints; Facilities layer shows helispots. |
| **Module 5 Comms Log** | Geofence alerts and segment assignments create CommsLog INFO lines. |
| **Module 2 Objectives** | Situation/Intel tab can embed current map snapshot into ICS-209 narrative. |

**Updates locked in for Module 6**

| **Decision** | | **How it’s now baked into the design** | |
| --- | --- | --- | --- |
| **POD model** | | Search-segment tables keep the classic SAR fields: probArea, probDetect, probSuccess = probArea × probDetect. Heat-map coloring and dashboard %-covered cards use that formula. | |
| **ADS-B integration** | | • New background service **adsb-ingest** subscribes to an ADS-B JSON feed (OpenSky, FlightAware Firehose, or your own RTL-SDR dump1090 on the ICP LAN). • Every position report inserts a TrackPoint with senderKind='ADS\_B' and the aircraft callsign in senderId. • Filter defaults to *altitude < 25 000 ft AND within 150 km* to keep the map uncluttered. | |
| **Auto-download tiles** | | • At incident start the Map service pulls MBTiles covering a **100 km radius from the ICP**. • When a new Task with geometry is created, a 50 km buffer around that task is queued for tile download if not already cached. • Tile jobs run in the background and retry when on Wi-Fi or LAN. | |
| **Setting** | **Where it lives** | | **What it does** |
| **ADS-B feed** ON / OFF | *Incident Settings → Mapping & Tracking* (visible to Comms Unit & Plans) | | • **ON** – launches the background **adsb-ingest** service, subscribes to your configured feed URL, and starts injecting TrackPoint rows with senderKind='ADS\_B'. • **OFF** – the ingest container stays idle; no ADS-B traffic touches the database. |
| **Feed URL / Port** | Global admin setting (DBAdmin) | | e.g. tcp://adsb-relay.local:30003 or OpenSky API key. |
| **Altitude & radius filters** | Same settings pane | | Defaults: *alt < 25 000 ft* **AND** *distance < 150 km* from ICP. |
| **Runtime badge** | Map toolbar shows a small ✈︎ badge (green when feed is running, grey when off). Users can long-press to jump to the nearest ADS-B target. | |  |

**Refinements locked-in for Module 6**

| **Topic** | **What changes** |
| --- | --- |
| **ADS-B ingest: only the aircraft you care about** | **Whitelist filter** is added.   **Where you configure it** **Incident Settings → Mapping & Tracking → ADS-B tab** (Comms Unit & Plans can edit).   **Two ways to load the filter**  1. **Auto-link to your AIR resources** – every row in *ResourceAircraft* that has a **tailNumber** (e.g., “N123AB”) or **callSign** (e.g., “Life Flight 3”) is copied into the whitelist at incident start.  2. **Manual add** – type a callsign, 24-bit ICAO hex (“A4B123”), or registration. Multiple entries allowed.   **Ingest logic**  • The adsb-ingest service still listens to the full feed, but **drops packets that don’t match the whitelist**.  • Matching is case-insensitive on callsign/registration; ICAO hex must be exact.  • You can toggle *“Accept new aircraft when they are added to the AIR roster”* to keep the list in sync. |
| **Map symbols: switch to the NAPSG Foundation Public-Safety Symbol Library** | **Default icon sheet** is now the *NAPSG-PS‐SL* SVG set (<https://www.napsgfoundation.org>).  • Categories used: **Operations**, **Search & Rescue**, **Medical & Hazard**, **Facilities & Infrastructure**.  • Icons are rendered at 24 px (vector scaling) with incident-status color halos (e.g., green complete, red needs-assist).  • MIL-STD-2525 remains an admin toggle but is *off* by default. |

## 6.2 ADS-B Ingest – revised data flow

mermaid

CopyEdit

graph LR

feed(ADS-B JSON feed)<-- all packets -->filter(Whitelist Filter)

filter-- only whitelisted callsigns -->track(Point senderKind='ADS\_B')

ResourceAIR-- tailNumber/callSign -->filter

* Turning **ADS-B ON** without any whitelist entries simply shows nothing until you add a callsign or AIR resource.
* Toggling **Off** stops ingest instantly (service polls the flag every 60 s).

## 6.3 UI tweaks

| **Screen** | **New elements** |
| --- | --- |
| ADS-B tab | • Whitelist list-box (callsign / ICAO / reg). • **Add From Resources** button populates from current AIR rows. • Toggle “Auto-sync with AIR resources”. |
| Map legend | NAPSG icons rendered; hover shows name (“SAR Team”, “Helispot”, etc.). |
| Trackpoint tooltip | For ADS-B symbols: Callsign, Altitude, Speed pulled from feed. |

### NAPSG Public-Safety Symbol Library – full coverage

* **All categories enabled:** Operations, Search & Rescue, Emergency Medical, Fire, Haz-Mat, Law Enforcement, Infrastructure, Natural Hazard, Weather, Utilities, Public Information, etc.—the entire current NAPSG SVG set.
* **Icon picker & legend:** the map’s layer-toggle/legend panel now lists every NAPSG class (collapsible groups so the UI stays tidy). When new icons appear in future NAPSG releases, they’ll auto-load after the next backend update.
* **Symbol assignment:** default mappings stay intuitive (e.g., SAR team → SAR icon, Helibase → Aviation icon). You can re-map any entity to any NAPSG symbol via **Admin ▸ Map Settings ▸ Icon Mapping**.
* **Exports:** snapshots, print maps, and the IAP inset all use the same NAPSG SVGs—so printed maps and digital views match.

Module 7 — **Reporting, Exports & Archiving**

(current design snapshot, reflecting every decision we’ve made so far)

| **#** | **Sub-module** | **Core entities / tables** | **What it does** |
| --- | --- | --- | --- |
| **7.1** | **PDF & Print Service** | FormInstance (id, periodId, formType, version, payloadJSON, signature, final) | Schema-first renderer auto-fills **all standard ICS forms** (201-232), plus any **custom forms** you create (check-lists, county templates, demob forms).  ▪ Typed-name + PIN digital signatures.  ▪ Multiple revisions; latest goes into every IAP snapshot until the period is closed.  ▪ Booklet composer joins 201-206-…-custom into one “IAP.pdf”. |
| **7.2** | **Form Registry & Custom-Form Manager** | FormType (code, title, schemaJSON, templateRef, includeInIAP) | • Admin GUI: Upload blank PDF/DOC, map fields, set validation, publish v1. • Versioning keeps old incidents on old templates. • “Checklist” field type supports critical items + digital signature. |
| **7.3** | **Bulk Exporter & Scheduled Dumps** | ExportJob (id, incidentId, type, schedule, targetURL) | • One-click CSV / JSON for any form type or table (e.g., Roster, CommsLog). • **ICS-309 exporter** — full CommsLog → NIFOG 309 PDF. • Scheduler (cron) can push nightly ZIPs to county share. |
| **7.4** | **After-Action & Analytics Dashboards** | MetricSnapshot (aggregated stats) | • Charts: hours per resource type, task completion %, comms volume, search POD plot. • Supports classic SAR POD math you selected. • Every snapshot stored with the incident for post-incident review. |
| **7.5** | **Incident Archive & Purge** | ArchiveRecord (id, incidentId, createdAt, sha256) | • One-click “Archive Incident” bundles **all PDFs + all JSON data + tile extracts + ADS-B logs** into a ZIP. • ZIP hash logged; ready for county retention rules. • Purge wizard removes non-required blobs after retention period. |

### Key workflows already wired

| **Workflow** | **How it works** |
| --- | --- |
| **IAP Snapshot** | Manual button **or** auto every 4 h. Captures latest version of every form flagged includeInIAP, plus map snapshot and the auto-generated per-unit ICS-214s. |
| **Automatic per-unit ICS-214** | Rows in **CommsLog** where (fromUnit or toUnit = unit) ∧ msgCategory ∈ {STATUS\_CHANGE, RETASKING, NEEDS\_ASSIST} feed the unit’s 214. Every period-close writes a fresh PDF per unit (no user steps). |
| **Custom demob checklists** | ResourceType → FormType link injects the right demob checklist (DEMOB-VEH, DEMOB-AIR, DEMOB-RAD, …). Signing the checklist flips status and stores the form as a locked PDF. |
| **ICS-205 / 205A + 217** | 205 builder pulls channel rows from **locked 217**; published 205 version is stamped & added to IAP booklet. |
| **ICS-309** | On demand, whole CommsLog dumps to FEMA/NIFOG 309 layout—includes From, To, Channel, Message. |
| **Analytics** | Dashboards pull from **MetricSnapshot** created at every IAP snapshot; POD heat-map from SearchSegment table. |

### Cross-module dependencies

* **Module 2** supplies Form Engine & signature widget.
* **Module 3** auto-creates status-change CommsLog lines → 214 exporter.
* **Module 5** provides filtered CommsLog rows, 205/217 data.
* **Module 6** map snapshot inserted as a page in the IAP and archived ZIP.
* **Module 4** demob checklists feed into PDF service.

### Configuration knobs (all live in Incident Settings ▸ Exports)

| **Setting** | **Default** | **Range / Notes** |
| --- | --- | --- |
| Auto-snapshot cadence | **4 h** | Off • 1 h • 2 h • 4 h • 6 h |
| IAP booklet order | FEMA default | Drag-list in Form Sequence Profile |
| 214 auto-export categories | STATUS\_CHANGE, RETASKING, NEEDS\_ASSIST | Multi-select from msgCategory ENUM |
| CSV dump schedule | Off | Cron picker; e.g., 03:00 local nightly |
| Archive retention | 7 years | Agency-wide default; per-incident override |

**Making the IAP form set selectable (Module 7.1 update)**

| **Aspect** | **How it will work** |
| --- | --- |
| **New “Build IAP” wizard** | When Plans (or IC) clicks **Publish IAP Snapshot** the wizard opens: 1️⃣ **Choose Forms** – a checklist of every FINAL form in the current period, grouped by type (201 → 206, 214 per-unit, custom, etc.).  • Mandatory forms (201, 202) are pre-checked and cannot be cleared. • Others are pre-checked per agency default but can be toggled off/on. 2️⃣ **Re-order** – drag to change sequence (defaults to FEMA order). 3️⃣ **Publish** – snapshot renders only the checked forms in the chosen order. |
| **Auto-snapshot behaviour** | • If the wizard has never been run for this period → falls back to legacy rule (*all forms where includeInIAP=true*). • If the wizard **has** been run once → auto-snapshots reuse the saved selection & order until the period closes. |
| **Persistent storage** | New table IapFormSelection sql<br>( snapshotId UUID, -- or periodId for “current”<br> formInstanceId UUID,<br> orderIndex SMALLINT,<br> PRIMARY KEY (snapshotId, formInstanceId) ) • When you publish manually, a row per selected form is written. • Auto-snapshot looks for the latest selection tied to the period. |
| **Agency-default template** | Admin page **Settings ▸ Exports ▸ Default IAP Forms** lets DBAdmin save a default checklist & order (drag-list). This populates the wizard the first time it opens. |
| **UI cues** | • Forms list tab shows a ⭐ on forms currently selected for the IAP. • Wizard remembers last state; “Publish” button is disabled until at least the mandatory set is satisfied. |
| **Audit trail** | AdminAuditLog records who changed the form selection and when; snapshot metadata stores the ordered list so anyone can verify what went into each IAP. |
| **Exports & archive** | Only the selected forms (plus map snapshot) are zipped into the snapshot PDF set and final archive. |
| **API additions** | GET /periods/{id}/iap-selection → returns current list and order. PUT /periods/{id}/iap-selection → saves the list (Plans/IC roles). |

**Outcome:** Plans and IC have full control over which forms appear in each IAP booklet—while mandatory forms remain enforced and auto-snapshots stay consistent with the last manual choice.

**System Module Outline (v0.1 Design)**

pgsql

CopyEdit

┌─ 1. Foundation & Core Services

│ ├─ 1.1 Authentication & RBAC

│ │ • OpenID-Connect JWTs, typed-name-PIN signature

│ │ • Org-chart roles auto-map to claims (IC, Ops, etc.)

│ ├─ 1.2 Offline-Sync Engine

│ │ • SQLite-CRDT on device ⇄ gRPC delta sync

│ │ • Edge-hub ↔ Cloud-hub replication, Line-A/Line-C geo libs pre-seeded

│ ├─ 1.3 File / Object Store

│ │ • MinIO-S3 buckets for PDFs, tiles, voice logs

│ └─ 1.4 Notification Service

│ • FCM/APNs/Win Toast + e-mail throttle

│

├─ 2. Planning Section

│ ├─ 2.1 Objectives Board (ICS-202 boxes 4-6)

│ ├─ 2.2 Dynamic Forms Engine

│ │ • FEMA 201-232 + custom JSON-schema renderer

│ ├─ 2.3 IAP Builder & Booklet

│ │ • \*\*Selectable form set\*\* wizard, order drag-list, auto snapshots

│ └─ 2.4 Situation / Intel (ICS-209, attachments)

│

├─ 3. Operations Section

│ ├─ 3.1 Task / Assignment Board (204 guts)

│ │ • Type-specific status flows, beacon cadence per type

│ ├─ 3.2 Field-report loop (auto STATUS\_CHANGE → CommsLog)

│ ├─ 3.3 Live Dashboards (task funnel, overdue, POD %)

│ └─ 3.4 Personnel / Resource link chips

│

├─ 4. Logistics & Resource Management

│ ├─ 4.1 Personnel Roster & Quals (T-card rack)

│ ├─ 4.2 Resource Libraries

│ │ • VEH, AIR, RAD, MED, TOOL, UAV, OTH tables

│ │ • Bar-code labels, scanner wizard

│ ├─ 4.3 Facilities & Ground Support (helispots, staging)

│ ├─ 4.4 Demobilization

│ │ • Asset-kind checklists (`DEMOB-VEH`, `DEMOB-AIR`, …)

│ └─ 4.5 Auto-form outputs (203, 204, 206, 219, 232)

│

├─ 5. Communications & Logs

│ ├─ 5.1 Radio / Message Log

│ │ • `msgCategory` filter, per-unit \*\*auto ICS-214\*\*, full ICS-309

│ ├─ 5.2 Threaded Chat (incident / period / task)

│ ├─ 5.3 Channel Library & Planning

│ │ • \*\*ICS-217\*\* with Mode A/D/M, multi-eligible-users, Line A/C geo lock

│ │ • 205/205A builder (pulls 217 rows)

│ └─ 5.4 ADS-B Ingest (whitelist on / off)

│

├─ 6. Mapping & Situational Awareness

│ ├─ 6.1 Offline Tile Cache (100 km ICP + 50 km/task)

│ ├─ 6.2 Live Tracking (tasks, personnel, ADS-B)

│ ├─ 6.3 Search-Planning (classic POD, segment polygons)

│ ├─ 6.4 Geofences & Alerts

│ └─ 6.5 NAPSG Symbol Map (all categories; MIL-STD-2525 toggle)

│

└─ 7. Reporting, Exports & Archiving

├─ 7.1 PDF / Print Service (schema-first, signatures)

├─ 7.2 Custom-Form Manager

├─ 7.3 Bulk Exporter & Scheduled dumps

├─ 7.4 Analytics Dashboards (hours, POD %, comms volume)

└─ 7.5 Archive & Purge (ZIP + SHA-256)

### Current ****Person**** record structure

(Module 4.1 – “Personnel Roster & Qualifications”)

| **Category** | **Field** | **Type / Format** | **Notes** |
| --- | --- | --- | --- |
| **Identity** | id | **UUID** (PK) | Device-generated when you create the record. |
|  | firstName | Text | — |
|  | lastName | Text | — |
|  | callSign | Text (optional) | Radio/voice ID shown on task board & Comms Log. |
|  | photoRef | File-blob UUID (optional) | JPEG/PNG; shown in drawer & T-card. |
| **Affiliation** | agency | Text | “County SAR”, “CAP-N234”, etc. |
|  | rankOrRole | Text (optional) | If your org tracks member rank. |
| **Contact** | phone | +1-555-555-1212 (optional) | Mobile / duty phone. |
|  | email | Text (optional) | For login & notifications. |
| **Operational status** | status | Enum: **AVAILABLE · EN\_ROUTE · ON\_SCENE · DEMOB · UNAVAILABLE** | Follows the roster status machine. |
|  | lastStatusAt | UTC timestamp | Auto-updates on any status flip. |
| **Barcode / Badge** (if enabled) | badge.barcode | Code 128 / QR string | Printed on ID card; scanner flips status. |
|  | badge.issueDate | Date (optional) | — |
|  | badge.expiryDate | Date (optional) | — |
| **Audit / system** | createdAt | UTC timestamp | — |
|  | createdBy | User UUID | — |
|  | updatedAt | UTC timestamp | — |
|  | updatedBy | User UUID | — |

**Module 4.1 — Personnel Roster & Qualifications**

*(fully updated with every addition we’ve locked-in)*

| **Theme** | **What’s in place** |
| --- | --- |
| **Identity & Affiliation** | • **Person.id** – *provided* unique ID (no auto-UUID). • firstName, lastName, callSign. • agency **and new** unitId → FK to **Unit** table (agency-scoped list of teams, squadrons, platoons). |
| **Contact** | phone, email, optional photoRef. |
| **Compliance** | • **Driver-licensed** hasDriversLicense BOOLEAN (toggle). • **Medical level** medCertLevel (enum: *FIRST\_AID, FIRST\_AID\_CPR, FR, EMT-B, EMT-A, EMT-P, RN, PA, MD, NP, Other*). |
| **Operational Status** | status enum → **AVAILABLE → EN\_ROUTE → ON\_SCENE → DEMOB → UNAVAILABLE** (customizable). lastStatusAt auto-timestamp. |
| **Qualifications** | Lookup tables **Qualification** & **PersonQual** (full CAP, NASAR, FEMA codes + extra list you imported). |
| **Availability Windows** | Table **PersonAvailability** (availFrom, availTo) – supports multiple future shifts; blank = always available. |
| **Badge / Bar-code** *(opt-in)* | badge.barcode, issueDate, expiryDate – Code128 printed; scanner flips status. |
| **Audit Trail** | RosterLog rows for every status change (manual or system-generated) — feeds analytics and the auto per-unit ICS-214. |

#### Related tables

* **PersonQual** – many-to-many link to **Qualification** (CAP, NASAR, FEMA codes you just imported).
* **RosterLog** – every status change (manual or system-generated) with who/when; feeds reports and the auto ICS-214.
* **TaskPersonnel** – zero-or-more rows when the person is dragged onto tasks; demob flips status back.

| **Change you asked for** | **How it’s baked into the schema & UI** |
| --- | --- |
| **1 . “Unit” field that is agency-specific** | • New table **Unit** sql<br>Unit(<br> id UUID PK,<br> agency TEXT, -- e.g. "CAP", "County SAR"<br> unitName TEXT -- "Sq 042", "Team B", "Platoon 3"<br>); • **Person.unitId** (FK → Unit.id). • **Roster form UI** – when you pick an **Agency**, the **Unit** dropdown is refreshed with that agency’s units (or shows “— none —”). • Bulk-import wizard: if a row’s unit isn’t in the table yet, it’s auto-created (flagged for DBAdmin review). |
| **2 . Person IDs will be provided, not auto-generated** | • Person.id now stored as **string / varchar(36)** (or UUID if your IDs are valid UUIDs) but **no default value**—the import or form must supply it. • Barcode/ID-card generator prints **exactly this provided ID** (no random UUID behind the scenes). • API rejects blank or duplicate IDs on create/import. |

CREATE TABLE Person (

id TEXT PRIMARY KEY, -- provided ID, not auto-UUID

firstName TEXT,

lastName TEXT,

callSign TEXT,

agency TEXT,

unitId UUID NULL REFERENCES Unit(id), -- new

phone TEXT,

email TEXT,

status TEXT CHECK (status IN

('AVAILABLE','EN\_ROUTE','ON\_SCENE','DEMOB','UNAVAILABLE')),

lastStatusAt TIMESTAMPTZ,

photoRef UUID NULL,

createdAt TIMESTAMPTZ DEFAULT now(),

createdBy UUID,

updatedAt TIMESTAMPTZ,

updatedBy UUID

);

**Personnel record — additional compliance & scheduling fields**

| **Change** | | **Implementation details** | |
| --- | --- | --- | --- |
| **1 . Driver’s-licence info** | | **New columns in Person** driversLicenseNum TEXT NULL, driversLicenseState TEXT NULL, driversLicenseExp DATE NULL *All optional. Validation:* state = 2-letter US/CA code; expiry ≥ today. | |
| **2 . Medical-licence info** | | **New columns** medLicenseNum TEXT NULL, medLicenseType TEXT NULL, -- “RN”, “EMT-P”, “MD”, etc. medLicenseExp DATE NULL Validation via lookup table of allowed licence types (DBAdmin editable). | |
| **3 . Availability windows** | | **Separate table**—supports multiple, repeating availabilities: sql CREATE TABLE PersonAvailability ( id UUID PK, personId TEXT REFERENCES Person(id), availFrom TIMESTAMPTZ, availTo TIMESTAMPTZ ); *Business rule:* open-ended windows allowed (NULL availTo = “until further notice”). | |
| **4 . UI / import changes** | | **Drawer** gets new “Compliance” section (driver/medical) and “Availability” tab (grid with Add / Edit). **Bulk CSV** accepts optional columns driversLicenseNum, driversLicenseState, driversLicenseExp, medLicenseNum, medLicenseType, medLicenseExp, availFrom, availTo. Unknown licence types flagged for review. | |
| **5 . Form & report surfaces** | | • Driver & medical data print on **203 Org Chart** (small icon) only for roles that need it (e.g., medical). • Availability windows influence the **Roster filter** (“Show people available tonight”). • Licence-expiry digest appears in the nightly alert e-mail if < 30 days remaining. | |
| **6 . Permissions & privacy** | | Only **Logs, Safety, Medical Unit** roles may view/edit licence fields. Availability windows editable by the member themselves in mobile profile. | |
| **Driver’s licence** | **hasDriversLicense BOOLEAN** (default **false**) | | • Roster drawer shows a simple toggle **“Driver-licensed ✔︎/✘”**. • Bulk CSV column hasDriversLicense accepts **Yes / No / 1 / 0**. |

|  |  |  |
| --- | --- | --- |
| **Medical licence / certification level** | **medCertLevel TEXT** (nullable) | FR (First Responder) · **FIRST\_AID** · **FIRST\_AID\_CPR** · EMT-B · EMT-A · EMT-P · RN · PA · MD · NP · Other • Selecting **Other** reveals a free-text box for the label. • Field is OPTIONAL—leave blank if the member has no medical credential. |

ALTER TABLE Person

ALTER COLUMN medCertLevel DROP CONSTRAINT IF EXISTS chk\_person\_med,

ADD CONSTRAINT chk\_person\_med

CHECK (

medCertLevel IS NULL OR

medCertLevel IN

('FR','FIRST\_AID','FIRST\_AID\_CPR',

'EMT-B','EMT-A','EMT-P','RN','PA','MD','NP','Other')

);

Person(

id TEXT PRIMARY KEY,

firstName TEXT, lastName TEXT,

callSign TEXT,

agency TEXT,

unitId UUID NULL REFERENCES Unit(id),

hasDriversLicense BOOLEAN NOT NULL DEFAULT false,

medCertLevel TEXT NULL

CHECK (medCertLevel IS NULL OR medCertLevel IN

('FIRST\_AID','FIRST\_AID\_CPR','FR','EMT-B','EMT-A','EMT-P',

'RN','PA','MD','NP','Other')),

phone TEXT, email TEXT,

status TEXT CHECK (status IN

('AVAILABLE','EN\_ROUTE','ON\_SCENE','DEMOB','UNAVAILABLE')),

lastStatusAt TIMESTAMPTZ,

photoRef UUID NULL,

createdAt TIMESTAMPTZ DEFAULT now(),

createdBy UUID, updatedAt TIMESTAMPTZ, updatedBy UUID

);

Unit(id UUID PK, agency TEXT, unitName TEXT);

PersonAvailability(id UUID PK, personId TEXT REFERENCES Person(id),

availFrom TIMESTAMPTZ, availTo TIMESTAMPTZ);

Qualification(code TEXT PK, name TEXT, issuer TEXT, renewPeriodMonths SMALLINT);

PersonQual(personId TEXT, qualCode TEXT,

issuedOn DATE, expiresOn DATE,

PRIMARY KEY(personId, qualCode));

RosterLog(id UUID PK, personId TEXT, oldStatus TEXT, newStatus TEXT,

ts TIMESTAMPTZ, changedBy UUID, note TEXT, systemGenerated BOOLEAN);

**Module 3 — Operations / Tasking**

*(all current design decisions rolled in)*

| **#** | **Sub-module** | **Core entities & fields** | **Highlights & workflow** |
| --- | --- | --- | --- |
| **3.1** | **Task record** | **Task** • id UUID • periodId FK • typeCode (e.g., SEARCH, LOG, MED …) • divisionCode (“DIV A”, “MED”, “STAGING”) • title, description • priority 1-5 • status (enum **per type**, see below) • Geo (geoType, geoData GeoJSON) • objectiveId? (link to ICS-202 objective) • createdBy, createdAt, updatedAt | One row per assignment that would normally appear on an ICS-204. |
| **3.2** | **Type-specific status flows** | **Lookup tables** TaskType(code, name, beaconIntervalSec) (defaults) TaskStatus(typeCode, code, label, orderIdx, isActive, beaconIntervalSec?) TaskTransition(typeCode, from, to) | *Examples* **SEARCH:** PLANNED → EN\_ROUTE → SEARCHING → REFITTING → COMPLETE (beacon 30 s when isActive=true). **LOG:** REQUESTED → IN\_PROGRESS → DELIVERED → COMPLETE (beacon 120 s). DBAdmin may add new types, states, transitions, and override beacon intervals (0 = suppressed). |
| **3.3** | **Live status & history** | **TaskStatusHistory** (id, taskId, newStatus, stamp, lat, lon, note, changedBy) | Drag-drop or quick buttons → FSM-checked transition → history row + GPS (if device location allowed). |
| **3.4** | **Personnel & resources bridge** | TaskPersonnel (taskId, personId, role, checkIn, checkOut)  TaskResource (taskId, resourceId, qty) | Drag chips onto the task card. Linking a person auto-flips their **Person.status → ASSIGNED**. |
| **3.5** | **Attachments & comments** | TaskComment (id, taskId, authorId, text, stamp) Photo uploads only (JPEG/PNG) stored in FileBlob. | No video, no audio (per your rule). |
| **3.6** | **Comms-Log linkage** | Every status change inserts CommsLog row with: • channel = "Status" • fromUnit = task.title / divisionCode • msgCategory = STATUS\_CHANGE • systemGenerated = true | Feeds the automatic per-unit **ICS-214**. |
| **3.7** | **Dashboards** | • Task-funnel widget counts by current status (columns vary with task type). • Resource-utilization bar (assigned vs. available). • Objective completion roll-up via objectiveId. |  |
| **3.8** | **Beacon engine** | Mobile SDK timer: interval = status.beaconIntervalSec ?? type.beaconIntervalSec. 0 s = no beacons. GPS breadcrumb stored as TrackPoint(senderKind='TASK'). |  |
| **3.9** | **API sketch** | GET /periods/{pid}/tasks?division=DIV%20A POST /tasks (create) PATCH /tasks/{id} (edit) POST /tasks/{id}/status {newStatus, note} POST /tasks/{id}/comment POST /tasks/{id}/attachments POST /tasks/{id}/personnel (add/remove list) |  |

| **Category** | **Code** | **Task-type name** | **Default flow →** | **Active states\*** | **Beacon sec** |
| --- | --- | --- | --- | --- | --- |
| **Search & Rescue – Ground / Water / Air** |  |  |  |  |  |
|  | SEARCH\_GROUND | Ground / K-9 search | PLANNED → EN\_ROUTE → SEARCHING → REFITTING → COMPLETE | EN\_ROUTE, SEARCHING | 30 |
|  | SEARCH\_URBAN | Urban / collapsed-bldg search | PLANNED → STAGED → CLEARING → COMPLETE | CLEARING | 30 |
|  | SEARCH\_HIGHANGLE | Rope / high-angle rescue | BRIEFED → RIGGING → WORKING → RTB → COMPLETE | RIGGING, WORKING | 20 |
|  | SEARCH\_WATER | Swift-/still-water search | BRIEFED → ON\_WATER → RECOVERING → COMPLETE | ON\_WATER | 45 |
|  | SEARCH\_CAVE | Cave / confined-space SAR | PREP → ENTERING → EXPLORING → EXITING → COMPLETE | ENTERING, EXPLORING | 30 |
|  | SEARCH\_AIR | Fixed-wing / helo SAR | BRIEFED → EN\_ROUTE → ON\_STATION → RTB → COMPLETE | EN\_ROUTE, ON\_STATION | 60 |
|  | SEARCH\_UAV | Drone imagery search | PREFLIGHT → LAUNCHED → ON\_TASK → RECOVERED → COMPLETE | LAUNCHED, ON\_TASK | 20 |
| **Medical & Evacuation** |  |  |  |  |  |
|  | MED\_EVAC | Casualty evacuation | DISPATCHED → PATIENT\_CONTACT → TRANSPORTING → COMPLETE | PATIENT\_CONTACT, TRANSPORTING | 45 |
|  | MED\_TREAT | Field treatment / triage | DISPATCHED → PATIENT\_CONTACT → COMPLETE | PATIENT\_CONTACT | 45 |
| **Logistics & Support** |  |  |  |  |  |
|  | LOG\_SUPPLY | General supply / logistics run | REQUESTED → IN\_PROGRESS → DELIVERED → COMPLETE | IN\_PROGRESS | 120 |
|  | LOG\_FOOD | Feeding / canteen ops | PREP → SERVING → CLEANUP → COMPLETE | SERVING | 180 |
|  | LOG\_WATER | Bulk water delivery | REQUESTED → FILLING → DELIVERED → COMPLETE | FILLING, DELIVERED | 120 |
|  | SHELTER\_MGMT | Shelter operations | SETUP → OPEN → OPERATIONAL → DEMOB → COMPLETE | OPERATIONAL | 300 |
| **Communications & IT** |  |  |  |  |  |
|  | COMM\_DEPLOY | Portable repeater / COML task | DEPLOY → ACTIVATED → TROUBLESHOOT → DEMOB → COMPLETE | ACTIVATED, TROUBLESHOOT | 90 |
|  | COMM\_NETOPS | Net-control operations | NET\_OPEN → HANDLING\_TRAFFIC → NET\_CLOSED | HANDLING\_TRAFFIC | 60 |
|  | COMM\_CHECK | System-wide comms check | INITIATED → CHECKING → COMPLETE | CHECKING | 300 |
| **Energy / Utilities** |  |  |  |  |  |
|  | ENERGY\_GEN | Generator deploy / fuel | DEPLOY → RUNNING → REFUEL → DEMOB → COMPLETE | RUNNING | 120 |
|  | UTILITY\_ASSESS | Power / gas inspection | DISPATCHED → SURVEYING → REPORT\_FILED → COMPLETE | SURVEYING | 90 |
| **Debris, Roads & Structures** |  |  |  |  |  |
|  | ROAD\_CLEAR | Road / debris removal | ASSIGNED → IN\_PROGRESS → COMPLETE | IN\_PROGRESS | 60 |
|  | DEBRIS\_SORT | Debris staging / sorting | ASSIGNED → SORTING → COMPLETE | SORTING | 90 |
|  | STRUCT\_STAB | Structure stabilization | ASSIGNED → SHORING → COMPLETE | SHORING | 60 |
| **Damage Assessment & Recon** |  |  |  |  |  |
|  | DAMAGE\_ASSESS | Rapid damage assessment | PLANNED → SURVEYING → REPORT\_FILED → COMPLETE | SURVEYING | 60 |
|  | FLOOD\_RECON | Flood-level reconnaissance | BRIEFED → ON\_WATER → RETURNING → COMPLETE | ON\_WATER | 45 |
|  | UAV\_RECON | UAV wide-area recon | PREFLIGHT → LAUNCHED → ON\_ROUTE → RECOVERED → COMPLETE | LAUNCHED, ON\_ROUTE | 25 |
| **Haz-Mat & Environmental** |  |  |  |  |  |
|  | HAZMAT\_MON | Haz-mat monitoring | DEPLOY → MONITORING → SAMPLE\_COLLECT → DEMOB → COMPLETE | MONITORING, SAMPLE\_COLLECT | 60 |
| **Fire** |  |  |  |  |  |
|  | FIRELINE\_CON | Hand-crew fireline | STAGED → CUTTING\_LINE → BACKHAUL → COMPLETE | CUTTING\_LINE | 30 |
| **Public Information & Liaison** |  |  |  |  |  |
|  | PIO\_MEDIA | Media / press briefing | DRAFTING → APPROVAL → RELEASED → COMPLETE | DRAFTING, RELEASED | 600 |
|  | LNO\_BRIEF | Agency liaison briefing | PREP → MEET\_AGENCY → ACTION\_ITEMS → COMPLETE | MEET\_AGENCY | 600 |
| **Safety & Security** |  |  |  |  |  |
|  | SAFETY\_AUDIT | Site safety inspection | SCHEDULED → INSPECTING → REPORT\_FILED → COMPLETE | INSPECTING | 300 |
|  | SEC\_PERIM | Perimeter / traffic control | POSTED → PATROLLING → HANDOFF → COMPLETE | PATROLLING | 60 |
| **Volunteer & Donations Management** |  |  |  |  |  |
|  | VOL\_MGMT | Volunteer registration | SETUP → REG\_OPEN → PROCESSING → DEMOB → COMPLETE | PROCESSING | 180 |
|  | DON\_MGMT | Donations management | SETUP → SORTING → DISTRIBUTING → COMPLETE | SORTING, DISTRIBUTING | 180 |
| **Air & Drone Logistics** |  |  |  |  |  |
|  | AIR\_DROP | Air-drop delivery | BRIEFED → EN\_ROUTE → ON\_RUN → RTB → COMPLETE | EN\_ROUTE, ON\_RUN | 45 |
| **ICS / Staff Work** |  |  |  |  |  |
|  | FIN\_ADMIN | Finance / Admin documentation | GATHER\_DOCS → REVIEW → SUBMIT → COMPLETE | REVIEW | 900 |
|  | PLANS\_DOC | Plan / Intel document prep | OUTLINE → WRITING → REVIEW → PUBLISHED → COMPLETE | WRITING | 900 |
|  | STAFF\_ICBRIEF | IC / Ops staff briefing | PREP → DELIVER → ACTION\_ITEMS → COMPLETE | DELIVER | 600 |
| Category → Air Search | Code | Task-type name | Default status flow → | “Active” beacon states\* | Beacon sec |
|  | SEARCH\_AIR\_FIXED | Fixed-wing visual search | PLANNED → BRIEFED → EN\_ROUTE → ON\_STATION → FIND → RETURNING → DEBRIEF → COMPLETE | EN\_ROUTE, ON\_STATION, RETURNING | 45 |
|  | SEARCH\_AIR\_HELO | Helicopter visual/hoist search | PLANNED → BRIEFED → EN\_ROUTE → ON\_STATION → FIND/HOIST → RETURNING → DEBRIEF → COMPLETE | EN\_ROUTE, ON\_STATION, FIND/HOIST | 30 |
|  | SEARCH\_AIR\_IMAGERY | Sensor / IR imagery search | PLANNED → BRIEFED → EN\_ROUTE → ON\_STATION → SENSOR\_PASS → RETURNING → DEBRIEF → COMPLETE | EN\_ROUTE, ON\_STATION, SENSOR\_PASS | 45 |
|  | SEARCH\_AIR\_UAV\_LR | Long-range UAV recon | PREFLIGHT → LAUNCHED → TRANSIT → ON\_TASK → RETURNING → RECOVERED → COMPLETE | LAUNCHED, TRANSIT, ON\_TASK | 60 |
|  | SEARCH\_AIR\_DROP\_RECON | Airdrop & rapid recon | PLANNED → BRIEFED → EN\_ROUTE → ON\_RUN → RETURNING → DEBRIEF → COMPLETE | EN\_ROUTE, ON\_RUN | 45 |
| Category → Air Search | Code | Task-type name | Default status flow → | “Active” beacon states\* | Beacon sec |
|  | AIR\_LINE\_SEARCH | Line (parallel‐track) search | PLANNED → BRIEFED → EN\_ROUTE → ON\_STATION → SEARCHING → RETURNING → DEBRIEF → COMPLETE | EN\_ROUTE, ON\_STATION, SEARCHING, RETURNING | 30 |
|  | AIR\_GRID\_SEARCH | Expanding-square / grid search | same as above | EN\_ROUTE, ON\_STATION, SEARCHING, RETURNING | 30 |
|  | AIR\_ELT | ELT / PLB homing | PLANNED → BRIEFED → EN\_ROUTE → ON\_STATION → HOMING → RETURNING → DEBRIEF → COMPLETE | EN\_ROUTE, ON\_STATION, HOMING, RETURNING | 30 |
|  | AIR\_IMAGERY | Aerial imagery (photo/IR) | PLANNED → BRIEFED → EN\_ROUTE → ON\_STATION → IMAGING → RETURNING → DEBRIEF → COMPLETE | EN\_ROUTE, ON\_STATION, IMAGING, RETURNING | 30 |