Survale iOS App Specification — v1.0

1) Overview

Survale is an iOS app (iPhone + iPad, iOS 18+) for small law-enforcement teams to coordinate live operations. It provides real-time location sharing (3–5s), an Apple Maps–centric UI with vehicle markers, operation-wide chat with media, after■action replay, and server-side PDF export. Tracks only while in active operations.

2) Functional Requirements

- Auth: email+password via Supabase; users belong to one primary team within an agency.
- Operations: Case Agent creates/starts/ends; invites; join requests; auto-end after 10 minutes of all-disconnected.
- Map: show teammates as vehicle icons with heading; staging & targets; nearby other-tenant ops as generic markers (tap to reveal CA).
- Chat: operation-wide (persisted 7 days post-end); DMs optional/ephemeral; photo/video attachments auto-compressed.
- Replay: smooth interpolation of trails; synchronized chat timeline; CA-only export to PDF.
- Notifications: invites & cross-tenant proximity (banner+sound/vibration); chat pushes only in background.
- Retention: purge all op data 7 days after end.

3) App Architecture

• Swift + SwiftUI, MVVM. • Networking via URLSession with async/await; Supabase client for Auth & Realtime. • State containers: OperationState, MapState, ChatState. • Background tasks: Core Location (Always Allow), BGProcessingTask for retry/cleanup. • Push: APNs; notification categories for invites, proximity, chat.

4) Navigation & Screens

- Login / Team Context: login; shows agency/team; switch profile vehicle/color before joining an op.
- Home (Operations): tabs for Active, Drafts, Ended; create operation (name, incident #), add targets, staging area.
- Map (Active Operation): vehicle markers; roster drawer (callsigns, status); targets/staging overlay; toggle trails (off by default).
- Chat: operation-wide channel; media picker; background push settings; @mentions (optional).
- Replay: time scrubber; speed controls; chat sync; export button (CA-only).
- Settings: haptics toggle; privacy text; sign out.

5) Map & Location Behavior

• Provider: MapKit (Apple Maps). Vehicle icons (sedan/SUV/pickup) tinted by user-selected color; heading rotates marker. • Updates: publish every 3–5s (foreground/background). Reduce noise via speed/accuracy thresholds; interpolate client-side for smoothness. • GPS-denied: rely on Core Location Wi■Fi/cell; show 'reduced accuracy' badge; maintain last-known position with muted style. • Trails: off by default; toggle shows last ~10 minutes of breadcrumbs.

6) Chat & Media

• Operation-wide chat persisted to Supabase; DMs via ephemeral WebSocket relay (no storage). • Media: HEIF/MP4 with automatic downscale (e.g., photos 2–3 MP; short videos ~720p). Progress UI and retry queue. • Read state: 'sent' + 'delivered' (read receipts optional later).

7) Notifications

• Categories: INVITE, PROXIMITY, CHAT. • Foreground: in-app banners for invites/proximity; chat stays silent. • Background: push for invites, proximity (banner+sound/vibration); chat badges only. • Deep links: open Operation detail; accept/decline invite from notification action.

8) Replay & Export

• Replay screen fetches locations_archive + op_messages; smooth interpolation; overlays targets/staging. • CA triggers server-side export; shows export status & download link; supports tagging time segments to include maps/coords.

9) Data Models (Client)

Core structs/classes: User, Team, Operation, OperationMember, Target{Person,Vehicle,Location}, StagingArea, Message, LocationPoint, ExportJob. Minimal DTOs mirror server contracts.

10) Networking Layer

- Supabase Auth (email/password).
- Realtime channels: op_{id}_locations and op_{id}_chat.
- REST to Node service: POST /v1/exports/:operationId/pdf, GET /v1/exports/:exportId.
- Uploads to Supabase Storage with signed URLs.

11) App Lifecycle & Background

• Request 'Always Allow' Location with full-screen rationale (tracking only during active operations). • Background fetch to retry failed uploads and send last-known heartbeat. • Auto-pause location publishes when op ends or user leaves the op.

12) Permissions & Privacy Copy

• Precise Location: “Survale uses your precise location *only* during active operations to coordinate your team.” • Notifications: “Used for operation invites, nearby-operation alerts, and chat updates when the app is not active.” • Media Library/Camera: “Used to capture and share photos/videos in operation chat.”

13) Settings

• Vehicle type & color (confirm on join). • Haptic alerts toggle for messaging. • Sign out.

14) iPad Layout

Split view: left = map; right = resizable rail with Roster/Chat/Targets tabs. Drag & drop photos into chat.

15) Accessibility

Support Dynamic Type; high-contrast pins; VoiceOver labels for markers (callsign, last update, accuracy); large tap targets.

16) Telemetry & Crash Reporting

Crash reporter allowed (no PII). Log only operational errors and anonymized performance metrics (latency, failed uploads).

17) Testing Plan (App)

• Unit tests: networking and DTO mapping. • UI tests: invite flow, join/leave, chat send/receive, replay scrub. • Field tests: dynamic/static surveillance, GPS-denied areas, proximity alerts, auto-end logic.

18) Build Config & Distribution

• Targets: Survale (Prod), Survale-Demo. • Configs: Development, Pilot, Production. • Enterprise distribution via MDM; TestFlight for pilot if needed. Bundle ID aligns with APNs topic.

19) Next Steps

- Create SwiftUI project scaffold (targets, dependencies, basic screens).
- Integrate Supabase Auth & Realtime; wire operation create/join flows.
- Implement Map screen (vehicle markers, overlays, trails toggle).
- Add Chat (operation-wide) with media upload and background push.
- Build Replay screen and export request flow.
- Harden background location behavior and reduced-accuracy UI states.