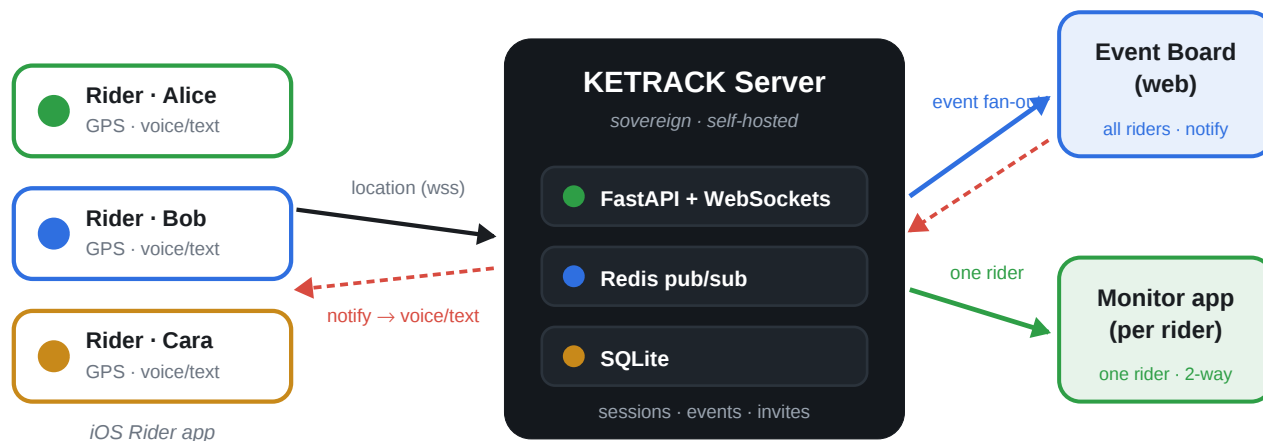


KETRACK — Phase 1 Capabilities

Events & the live Board — what the current build (server v0.4.0, Rider v0.4.0) does today, by role. Generic real-time entity telemetry: track people or assets in motion, watch them live, and message them — self-hosted, no third-party cloud.

The system at a glance



Riders stream position to a self-hosted server; a per-rider Monitor and an event-wide Board observe, and messages flow back to riders as voice/text.

Capabilities by role

Rider — the tracked entity (iOS)

The person or asset in motion. New in v0.4.0: name + event code.

- Live GPS on a map with the route drawn as a trail; follow-me toggle.
- Live metrics: speed, distance, elapsed time, and a LIVE/OFF status dot.
- Start / Stop, plus Pause / Resume (freezes time, distance, broadcasting).
- **Name + optional event code** at start (remembered) — join an event to appear on the Board.
- Two privacy modes: **Global** (shareable watch code) or **Private** (named, revocable invites).
- Hears inbound messages from a Monitor or the organizer — banner + spoken aloud (TTS).
- On-device ride history (GPX): review, map, stats, delete, share/export.
- Resilient: auto-reconnect with backoff; buffers ~2 min of location offline and flushes on reconnect.

Monitor — the 1-to-1 watcher (iOS)

Follows a single rider by code. Unchanged in v0.4.0 and still works.

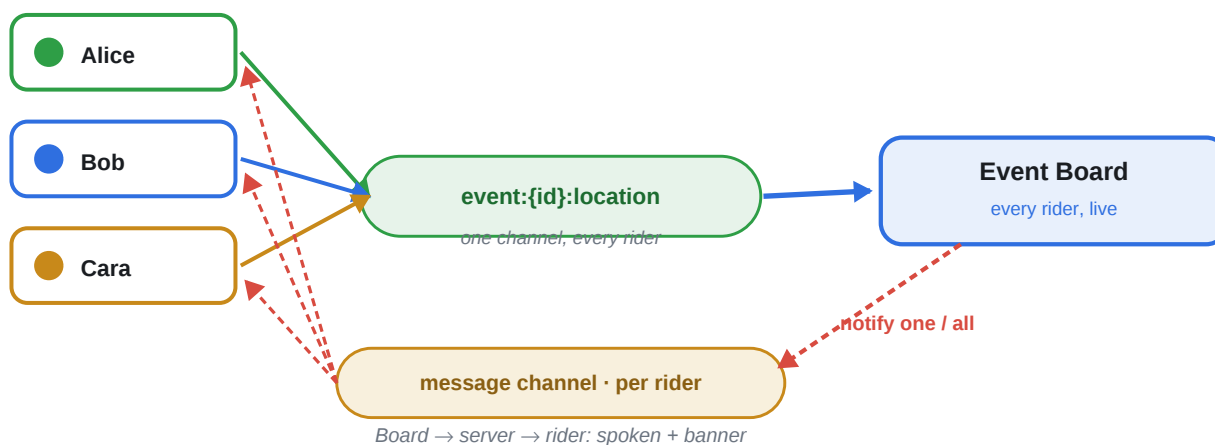
- Pair by a Global watch code or a Private invite code.
- Live view of that one rider: position + accumulating route, name, speed / distance / time.
- Two-way text messaging to the rider (rate-limited), spoken aloud on their phone.
- Clean termination with a reason if the ride ends, access is revoked, or the token is bad.

Board — the event observer (web)

The 1-to-many command view. New in v0.4.0 — the Phase 1 headline.

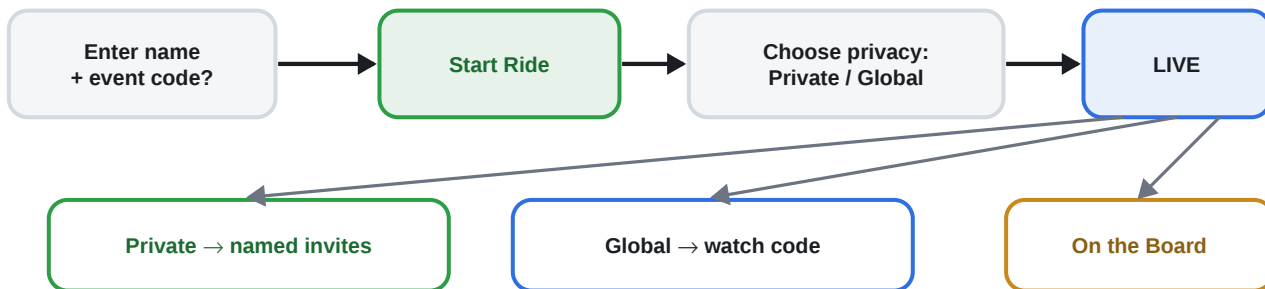
- Connect with server host + token + event code (secure toggle); settings remembered.
- **Every rider in the event on one map**, each a named marker with a trail; auto-fit.
- Roster: name, speed, distance, elapsed, and freshness — amber when stale, grey when offline.
- Click a rider to pan to them.
- **Notify one rider or all at once** over WebSocket, with a delivery acknowledgement.
- Auto-reconnects if the connection drops.

The new mechanism — event fan-out & notify



Each rider's frames publish to one shared event channel the Board reads; the Board's notify routes back through each rider's own message channel.

A ride, start to finish



Pause / Resume · Stop ends the session

Event membership is independent of privacy mode — a rider can be on the Board and watched individually at the same time.

Prove it in six steps

- | | | |
|---|-----------------|---|
| 1 | Create an event | POST /api/v1/events → note the event code. |
| 2 | Open the Board | event-board.html → host + token + code → Connect. |

3	Rider 1 joins	Rider app: name + event code → Start. Appears live.
4	Rider 2 joins	Second device, same code → two riders on one Board.
5	Notify	Board → one rider or all → banner + voice on the phone(s).
6	Legacy check	Monitor app pairs by watch code — 1-to-1 still works.

Full click-by-click detail with curl commands and edge cases: [Docs/TEST-SCRIPT_Phase1_020826.md](#).

Not yet in this build: mesh/off-grid transport (assumes internet/LAN today), full accounts, and the live event-name confirmation on the rider's code field. The Rider app is written but needs an Xcode build before field use.